JSTOR

WHITE PAPER

Digitizing Printed Arabic Journals:

Is a Scalable Solution Possible?

NEH Award PW-253861-17
JSTOR Arabic-Language Digitization Planning
March 29, 2019

John Kiplinger
Anne Ray

In 2017, JSTOR received a grant from the National Endowment for the Humanities to determine the feasibility of developing a scalable solution to the digitization of Arabic-language texts. Throughout the two-year project, we worked in deep collaboration with other organizations and initiatives in the academic community. This white paper is intended to share the result of this investigation as broadly as possible to help inform and further the work of others. We hope that individuals and organizations working in the areas of broadening access to Arabic language material – through digitization, OCR, machine learning, search and more – will read this and share their comments and experiences.

The JSTOR Arabic-Language Digitization Planning project has been made possible in part by the National Endowment for the Humanities: Exploring the human endeavor.

Any views, findings, conclusions, or recommendations expressed in this paper, do not necessarily represent those of the National Endowment for the Humanities.

JSTOR, a service of the not-for-profit organization ITHAKA, collaborates with the academic community to help libraries connect students and faculty to vital content while lowering costs and increasing shelf space, provides independent researchers with free and low-cost access to scholarship, and helps publishers reach new audiences and preserve their content for future generations.

ITHAKA is interested in disseminating this paper as widely as possible. Please contact us with any questions about using the report: support@jstor.org.

JSTOR also gratefully acknowledges the contributions and cooperation of the following:

1. The OpenITI team, particularly Matthew Miller and Benjamin Kiessling, for providing significant knowledge, time, and effort related to Arabic OCR that was essential to the successful outcome of this project.
2. Amelie Beyhum and Lokman Meho at American University in Beirut, publisher of the journal *al-Abhath*, for their collaboration in allowing ITHAKA to use this journal in our investigation and for approving the use of the already-created page images from Project AMEEL for vols. 1-50/51.
3. The Yale University Library for maintaining the page images created for Project AMEEL and supplying copies of those images for *al-Abhath* to ITHAKA. Specifically, we thank: Susan Gibbons, University Librarian and Deputy Provost, Collections & Scholarly Communication, Yale University; George Ouellette, Digital Collections and Repository Manager; and Elizabeth Beaudin, Manager, International Digital Projects.
4. The Harvard University Library/Digital Library of the Eastern Mediterranean for hosting an Arabic OCR meeting in June 2017 that introduced ITHAKA staff to the OpenITI team. Specifically, we thank: Sarah Thomas, Vice President for the Harvard Library and University Librarian, Roy E. Larsen Librarian for the Faculty of Arts and Sciences, Harvard University; and Kathryn Schwartz, Assistant Professor of History, University of Massachusetts, Amherst.
5. Apex CoVantage, a long time digitization service provider for ITHAKA, for providing digitization and workflow expertise and for discussion of developing a large scale workflow for Arabic journals.
6. Guy Barak, the Middle Eastern and Islamic Studies Librarian at NYU Library, for his generous advice and support.
7. The members of MELA who have provided input into JSTOR's efforts, especially William Kopycki, Robin Doherty, and Marlis Saleh.

Summary

In 2017, JSTOR received a grant from the National Endowment for the Humanities to investigate processes for digitizing Arabic-language scholarly content. Our goal in the project was to develop a workflow for scanning Arabic materials--especially journals--that is reasonably cost-efficient, feasible to implement at scale, and likely to produce high-quality images and metadata, including fully searchable text.

Our investigation involved three components:

1. Licensing the rights needed to digitize a sample Arabic-language journal, which we were able to accomplish thanks to the generous participation of the American University of Beirut for their journal *al-Abhath*.

2. Creating a set of metadata capture guidelines for Arabic-language journals. These guidelines are important because accurate metadata is an essential factor in content discovery. Guidelines are used to ensure consistent capture and naming of elements across the issues of a journal and across many journals. While in most ways JSTOR's standard metadata capture guidelines for journals would suffice, there are some aspects to Arabic metadata capture that require specialized direction and on which we provide details later in this paper.

3. Digitizing a small set of Arabic-language journal content. This component included not only scanning the journal pages, but finding an OCR (Optical Character Recognition) solution that provides consistently highly-accurate output, something that is not yet yielded via commercially available OCR packages. As detailed later in this paper, JSTOR was fortunate to have an opportunity to work with the Open Islamicate Texts Initiative (OpenITI)[1] which had seen very promising results using open source software called Kraken that their technical lead, Benjamin Kiessling, developed.[2] JSTOR was able to conclude this component of the project working with OpenITI and Apex CoVantage (Apex), a long-term digitization service provider for JSTOR.

Through the OpenITI, JSTOR, and Apex collaboration, a workflow was created that can be summarized through these steps:

- OpenITI reviewed pages from the journal *al-Abhath* for distinct typefaces and created training data for its OCR software representative of those typefaces.
- OpenITI ran OCR on pages from issues of *al-Abhath* selected by JSTOR, and assessed the accuracy of that output.

[1] For more information on OpenITI, visit their site at: https://iti-corpus.github.io/index.html.

[2] "Important New Developments in Arabographic Optical Character Recognition (OCR)" available at: https://www.academia.edu/28923960/Important_New_Developments_in_Arabographic_Optical_Character_Recognition_OCR_

- Apex integrated this OCR output into its workflow and provided JSTOR with final deliverables conforming to JSTOR-determined specifications.
- JSTOR provided feedback and direction on Apex's implementation of the metadata specifications.

In this portion of the project, JSTOR's primary role was to coordinate communications and handoffs with/between OpenITI and Apex, set and monitor deadlines, and assess how and if what was learned by this investigative project could translate into a cost-effective and operationalized workflow for digitizing Arabic journal content at a high level of accuracy.

Through this investigation, we concluded that, using new metadata guidelines and OpenITI's software, and leveraging specific workflows created jointly with Apex, it is possible for JSTOR to digitize Arabic language journals with the high-degree of accuracy needed to support search and discovery at a cost of approximately $3 per page, with the promise that this per page cost could be reduced further through continuous improvements in the OCR software engine.

In this white paper, we contextualize our investigation in the broader landscape of digital scholarly literature in Arabic. We then document our approach and findings from this project, which took place over 20 months from April 2017 through December 2018. And finally, we lay out some areas we identified for potential further research.

Introduction: Reflecting Global Scholarship in our Digital Scholarly Resources

The ease of digitizing and making available online content in English and other languages that use the Latin character set has obscured a much more complicated situation for content using non-Latin character sets. Important scholarly texts in non-Latin languages are often still available only in print form, or have been digitized with lower quality OCR or no OCR at all, leaving them largely undiscoverable online to scholarly audiences.

Arabic-language scholarly texts collectively form a prime example of the challenges of digitizing texts that use non-Latin character sets. While there have been several valuable efforts to digitize Arabic-language scholarly journals and special collections, a lack of highly accurate OCR software for Arabic as well as the costs for granular metadata capture has hampered both the sustained digitization of printed texts, and the discoverability and usability of those Arabic-language texts that have been converted from print to digital form. Overcoming these challenges is crucial not only to ensure that our understanding of scholarship is global, but because Arabic-language scholarly texts face special challenges around access and preservation. Addressing a key prerequisite for making digitized texts searchable and usable—an effective and cost-efficient digitization and OCR process—could be an important accelerator in a scholarly community-led effort to digitize and digitally preserve these at-risk materials.

The need for Arabic language scholarly materials online is well established.[3] In participating in this project, JSTOR hoped to be a support to the educators, librarians, and scholars already engaged in in this area. A selection of these efforts is described here:

- Project AMEEL, a project from 2005-2007 based at Yale University, digitized 350,000 pages of Arabic content.[4]
- The wide-ranging work of the Middle East Librarians Association has tackled many aspects of the preservation, digitization, and collecting of materials in Arabic across geography and content type. MELA members have built infrastructure in numerous areas, from cataloging to discovery, and is a leader in the preservation of cultural heritage and in working to address the urgent problems of looting and destruction of materials in areas of the region affected by conflict. Directly related to this project is Evyn Korpf's essential foregrounding of the challenges of Arabic OCR, published in 2008.[5]
- Arabic Collections Online, sponsored by New York University in Abu Dhabi, contains over 10,000 volumes in Arabic, digitized to a rigorous and well-documented standard, and has "a goal of reaching 20,000 volumes."[6]
- The Digital Library of the Middle East, founded by The Council on Library and Information Resources, focuses on cultural heritage materials on an open source platform developed by Stanford University Libraries, with capabilities for display in both Romanized and Arabic forms.[7]

[3] The need for scholarly materials online in other languages that use the Arabic alphabet, such as Persian or Urdu, is also well established, but for the purposes of defining scope, our project concerned Arabic only.

[4] For more information about AMEEL and access to available content, visit: https://web.library.yale.edu/digital-collections/arabic-and-middle-eastern-electronic-library. For a detailed review of AMEEL, see also: Samoeil, Simon. "Digitization of Near East Materials From a Curatorial Point of View." *MELA Notes*, no. 83, 2010, pp. 39–41. JSTOR, https://www.jstor.org/stable/29785925.

[5] Kropf, Evyn. "Training Challenges: A Practical Report of an Arabic OCR Experience." *MELA Notes*, no. 81, 2008, pp. 1–13. JSTOR, https://www.jstor.org/stable/29785882. See also Kropf, Evyn, and Jonathan Rodgers. "Collaboration in Cataloguing: Islamic Manuscripts at Michigan." *MELA Notes*, no. 82, 2009, pp. 17–29. JSTOR, https://www.jstor.org/stable/29785905, and Moustafa, Laila Hussein. "The Role of Middle East Studies Librarians in Preserving Cultural Heritage Materials." *MELA Notes*, no. 90, 2017, pp. 15–22. JSTOR, https://www.jstor.org/stable/26407383.

[6] Parrott, Justin. "Crossing Boundaries with Arabic Collections Online." *MELA Notes*, no. 90, 2017, pp. 13–14. JSTOR, https://www.jstor.org/stable/26407382. See also http://dlib.nyu.edu/aco/.

[7] https://dlme.clir.org/; https://dlme.clir.org/2018/01/31/announcing-digital-library-middle-east-prototype/, accessed 27 Feb 2019.

- Commercial endeavors such as Gale Cengage and al-Menhal have also produced substantial digitized collections.[8]

Our aim in this project was to investigate methods for digitizing Arabic-language scholarly journal content at scale such that the output was highly accurate and would reflect best practices, while also being cost-effective. The intent was for the outcomes of this project to be helpful in guiding JSTOR's own potential plans to digitize Arabic-language texts, but also to be informative to others. By fully documenting our process, we hope our work can benefit future digitization efforts of Arabic-language scholarly texts in the broader academic community, as well as inform the possible re-digitization of work undertaken five or ten years ago to improve its discoverability. Ultimately, we hope that our contribution to the scholarly community's collective knowledge base in this area can help to stimulate further interest in the digitization of Arabic-language scholarly materials, and thus increase the volume of Arabic-language content that is preserved in digital form and made available for research and teaching.

Methodology

JSTOR was founded to create a shared digital library of printed materials to preserve this content centrally on behalf of libraries and to make it widely accessible online. JSTOR launched in 1997 by digitizing the complete archival back runs of ten economics and ten history journals. Today, JSTOR 's Archive Collections contain the complete back runs of over 2,700 journals totaling more than 75 million pages, across over 75 disciplines. The context for our exploration into Arabic is this interdisciplinary corpus, and stems from a broad effort to expand globally, focusing on academic and cultural journals in the humanities and social sciences, where the focus of this corpus has been. As the JSTOR collections have expanded, we have aimed to digitize and make available journals and other materials to better represent global scholarship, through greater diversity of language and geographic coverage.

JSTOR's approach to building its corpus of scholarly materials is selective in nature. Publications that are invited to participate have a long history, are well-established in their fields, are rigorously reviewed by expert peer groups, and have made a long-standing and ongoing contribution to intellectual life. Additionally, JSTOR works with advisors, including subject specialist librarians and faculty,[9] who identify publications that represent sources of irreducible value in their respective fields. In keeping with this approach, working with an Arabic-language journal with a similar standing was a key

[8] https://www.gale.com/intl/primary-sources/early-arabic-printed-books-from-the-british-library. https://www.almanhal.com/en/Collection/JournalCollections

[9] Examples of advisory groups convened for JSTOR collection development include: https://about.jstor.org/whats-in-jstor/security-studies/ and https://about.jstor.org/whats-in-jstor/lives-of-literature/

choice. In the case of Arabic language scholarship, JSTOR was fortunate to have the expert advice of MELA, with whom we worked in 2016-7 to conduct an informal survey on publications of high value in Arabic for which a need for digital access was clear. Additionally, while some of the projects mentioned above have focused on out-of-copyright materials like books (for which a critical need also exists), this project would focus on a serial in the humanities and social sciences with both a long history and highly regarded issues regularly produced today.

This approach led us to the journal *al-Abhath*, a preeminent Arabic-language scholarly journal in the humanities published by the American University of Beirut (AUB). *Al-Abhath* has been published at AUB since 1948 and not only captured a critical period in the cultural and political history of the contemporary Middle East, but also represents the vibrant dialogue among scholars in Lebanon and globally. With a loose focus on the study of the Arab world, the journal publishes articles from scholars working across the humanities and social sciences. *Al-Abhath* is similar to other journals from around the world that have established themselves as irreducible in their value, such as *Daedalus* in the United States or *Revue des Deux Mondes* in France. It is also multilingual; its primary language of publication is Arabic, but it also includes some articles in other languages, as well as English-language abstracts and tables of contents. Its long influence, broad approach, and the presence of multiple languages made this title an especially appealing journal with which to begin our investigation.

Fortunately, a portion of *al-Abhath* had earlier been included in Project AMEEL, mentioned above. AMEEL provides access to this journal at the issue level and with limited searchability. Building on this work, JSTOR's project would look at both the potential for producing consistently highly accurate OCR (i.e., high 90s for character-level accuracy on a per page basis) and more granular metadata. AUB generously gave Yale University Library permission to provide JSTOR the page images of *al-Abhath* created for AMEEL for use in this project. With the digital page scans in hand, our focused turned to OCR and metadata capture.

JSTOR, has two decades of experience in managing large-scale text digitization projects, both in Western languages and otherwise. Presently journal articles in approximately 30 languages appear on JSTOR, though very few are in non-Latin character set languages. Notably, in 2012-2015, JSTOR partnered with the University of Haifa and the National Library of Israel to conduct a successful grant-funded Hebrew-language journal digitization project.[10] While the building of this corpus has yielded some expertise in multilingual content among our own staff, we also have long-term working relationships with two main digitization service providers (Apex and Ninestars Information Technologies) whose own experience and expertise has amply supported JSTOR's needs. JSTOR worked with Apex on the Hebrew-language digitization project, and their

[10] More about this collaboration can be found at: https://lib.haifa.ac.il/index.php/en/projects-collections-eng/arch-proj-eng/ijstor-about-eng

experience from this project, both with JSTOR's requirements and with developing workflows to process this content accurately and at scale, led us to invite them to work on this project as well.

While libraries frequently do manual capture of each piece of metadata, both of JSTOR's digitization service providers have workflows and systems set up to use the OCR output as the basis for metadata (where appropriate) with manual cleanup done as needed. This workflow creates efficiencies but also requires highly accurate OCR output in order to support those efficiencies. As noted elsewhere in this paper, highly accurate OCR is generally feasible for Western European languages using the Latin character set, but it is known to be more problematic for Arabic. Using ABBYY FineReader, a widely used commercial OCR software, we saw per page character accuracy rates in the 70-75% range (as opposed to 99% or higher for Latin character OCR). While we expect this accuracy to increase as ABBYY makes improvements, we were looking for an accuracy rate in the high 90s in order to maintain these efficiencies in metadata capture as well as support the end user discovery experience.

In June 2017, at Harvard University's invitation, JSTOR attended a meeting sponsored by Harvard's Digital Library of the Eastern Mediterranean. The focus of the meeting was Arabic OCR, and among the attendees were the Principal Investigators (PIs) for OpenITI. OpenITI reported utilizing a different technical approach that could consistently yield page-level character accuracy rates of 97% or higher. Of particular interest was their desire to create a more robust, accurate, and cost-effective OCR resource for scholars of Arabic and Persian texts using open source OCR software called Kraken, developed by the team's technical lead Benjamin Kiessling. Their focus on scholarly communication aligned with JSTOR's mission, and JSTOR therefore initiated discussions with the OpenITI PIs on the grant project. Our primary contact was Matthew Miller.

A crucial factor in the accuracy of the OpenITI's OCR output is the training of the Kraken software. While OpenITI's 2016 paper and Appendix A of this paper describe the training process in more detail and should be read to get the most informed understanding, we will provide a brief overview:

- First, the content to be processed must be reviewed to identify different typefaces and variants within those typefaces.
- An appropriate volume of lines of text representing each typeface and its variants are selected.
- A machine-readable version of each line is manually captured and associated with the digital image of the corresponding text. Those images and the corresponding machine-readable version are then used to train Kraken to recognize a flat image of similar text in the future. The greater the number of lines of text that are captured for training purposes, the more likely that the

variations in the typeface will be adequately represented in the training data, such that permutations of the text can be subsequently recognized by Kraken.

- In time, as Kraken is trained on more typefaces, a more robust generalized recognition model for character recognition emerges that yields greater accuracy *across* typefaces. Furthermore, using this generalized model, as additional typefaces are identified, it is anticipated that less and less training data will need to be created in order to attain the desired high accuracy rates. As implied above, a further factor in Kraken's success is its use of a neural network to recognize characters within a line of text (i.e., as they would normally appear) rather than as standalone characters.

Discussions between JSTOR and OpenITI resulted in an initial agreement in December 2017 to identify the typefaces present in the page images of *al-Abhath*. These issues covered vols. 1-50/51 (1948-2001/2002), with some gaps. OpenITI would create training data based on the identified typefaces and use the data to train Kraken. Next, it would run OCR on pages for a subset of *al-Abhath* issues selected by JSTOR. Those page images and corresponding OCR files would then be integrated into a conversion process conducted by Apex.

OpenITI's review of the available *al-Abhath* page images indicated that there were two main typefaces with multiple variants within each. It was decided that a total of 7,000 lines of text would be sufficient to create the needed training data, and OpenITI labored during the first quarter of 2018 on the manual capture of that data. Upon completion of the training data in early April, it was used via fully automated processes to train the OCR software.

During the second and third quarters of 2018, JSTOR worked with OpenITI and Apex on OCR output, first producing a sample and then later a full set of OCR, for review and related questions. In order better to integrate the OCR output into its own workflows, Apex requested that OpenITI run OCR on zone-level images rather than full-page images. Roughly speaking, zone-level images are created by dividing images of the digitized pages into zones to specify reading order. A single page may have multiple zones depending upon the complexity of its formatting. In its own workflows for JSTOR, Apex runs OCR on zone-level images and then reassembles the OCR zones into a single page-level file for delivery to JSTOR. Consequently, there was discussion of zone-level vs. full-page images as input in order to ensure a common understanding of inputs as well as outputs. Apex raised multiple additional technical questions once they saw OCR output in order to confirm their understanding of it. While it is not necessary to detail these questions, it is worth noting their existence, as any such integration of new, external output into an already established workflow will raise numerous questions and requests for clarification, and time must be allotted to address them.

In July 2018, Apex took four *al-Abhath* pages for which OpenITI had provided OCR output and conducted a very small test to compare the accuracy of the OpenITI output with that of ABBYY. The results of this comparison are described in detail in Appendix A. In brief, the OpenITI output was, with some exceptions, notably more accurate than that of ABBYY, attaining the desired accuracy rates in the high 90s. Given the *very* limited nature of this test, it was not meant to provide a statistically meaningful measurement of average accuracy for the two OCR applications. Instead, it confirmed our earlier understanding of their *relative* accuracy and underscored the promising nature (from an accuracy standpoint) of OpenITI's approach. OpenITI subsequently reviewed Apex's findings in more depth and determined that there were different understandings of what constituted an error (e.g., conversion of Eastern Arabic numerals in the source text into Western Arabic numerals in the OCR output, exclusion of diacritics from the OCR output). There were also a smaller number of instances where there was no error at all. Given OpenITI's understanding of Kraken, its output, and the Arabic character system in general, JSTOR asked them to undertake an accuracy assessment of Kraken's output across a wider number of *al-Abhath* pages. OpenITI agreed and spent the fourth quarter of 2018 conducting a detailed assessment on fifty pages, the results of which are noted below as well as being described in detail in Appendix A.

While OCR was a crucial element in the investigation into Arabic digitization, metadata capture is of course also very important. Metadata capture was divided into two main efforts: 1) A JSTOR effort to develop metadata capture guidelines, and 2) Apex's work to capture metadata from the *al-Abhath* issues selected for digitization.

During the fourth quarter of 2017, a metadata librarian from JSTOR's Content Management team worked with an Arabic-language consultant already familiar with our standard metadata requirements. JSTOR is in the process of migrating from our current journals metadata specification (i.e., an adaptation of the National Library of Medicine's Archiving and Interchange Tag Suite, or NLM DTD[11]) to a JSTOR-specific adaptation of the NISO Journal Article Tag Suite, or JATS[12], a successor to the NLM DTD. We determined that developing a full set of distinct metadata capture guidelines for Arabic-language journal content was not needed. Arabic-language scholarly journals do not generally differ significantly in the presence, location, or formatting of relevant metadata from their Western language counterparts. Instead, JSTOR's metadata guidelines for Arabic journals addressed areas such as, but not limited to:

- Ensuring a consistent understanding and application of how Arabic-language metadata (e.g., author names) were to be parsed into relevant constituent elements.

[11] More information about the NLM DTD can be found at: https://dtd.nlm.nih.gov/.

[12] More information about JATS can be found at: https://jats.nlm.nih.gov/.

- Ensuring clarity regarding the capture of metadata present in multiple languages in the source (e.g., capturing an article title when it is given in both Arabic and English, or ordering pages in an issue containing both Arabic- and English-language sections, where the two sections start at opposite ends of the issue and progress inward).
- Ensuring that JSTOR practices (e.g., collecting non-article pages at the front of the issue into an article called "Front Matter") had appropriate Arabic-language counterparts.
- Accounting for JSTOR system requirements that might not be supported by Arabic-based metadata. For example, JSTOR's systems require a publication date that reflects the Gregorian calendar. In cases where the available publication date reflects the Islamic/Hijri calendar, conversion of the date is necessary. The metadata guidelines provide direction on doing this.

The result was that while JSTOR expected that Apex would apply our standard JATS-based metadata capture guidelines to any content in this project, a *companion* document covering metadata capture specific to Arabic journals content was developed. This companion document, while applied only on *al-Abhath* for this project, can be used for any Arabic journal content that JSTOR processes as well as informing metadata capture practices for other Arabic journal digitization projects. The companion document is attached as Appendix B.

In the fourth quarter of 2018, Apex processed the page images and corresponding OCR output for the selected whole issues of *al-Abhath* totaling 4,000 pages.[13] These issues were selected because their content represented a variety of metadata capture challenges. Apex submitted final deliverables by the end of the quarter.

Findings

In conducting this project, we found that it is possible to attain highly accurate OCR output utilizing the method of a review of the source content for distinct typefaces, the creation of training data representing those typefaces, and then the training of Kraken using that data. Not only did the resulting typeface-specific recognition models provide improved accuracy, Kraken's generalized recognition model covering all the typefaces known to it can generate character accuracy rates of **98%** or higher. Such rates make discovery via full text searching more plausible and consistent and can make metadata capture more accurate and economical.

[13] The issues selected for processing were: vol. 1, no. 2 (June 1948); vol. 4, no. 4 (Dec. 1951); vol. 8, no. 3 (Sept. 1955); vol. 8, no. 4 (Dec. 1955); vol. 11, no. 3 (Sept. 1958); vol. 11, no. 4 (Dec. 1958); vol. 16, no. 1 (Mar. 1963); vol. 16, no. 2 (June 1963); vol. 21, no. 1 (Mar. 1968); vol. 21, no. 2/4 (Dec. 1968); vol. 22, no. 3/4 (Dec. 1969); vol. 23, no. 1/4 (Dec. 1970); vol. 27 (1978-1979); vol. 31 (1983); vol. 33 (1985); vol. 34 (1986); vol. 36 (1988); vol. 39 (1991); vol. 45 (1997); vol. 46 (1998); vol. 48/49 (2000/2001).

The project also revealed a set of challenges alongside these positive results. Some of these challenges are technical in nature, while some relate to the need to scale. Some of the technical challenges represent obstacles that must be addressed in order to make digitization at scale with highly accurate results a practical possibility. Among these are consistent formatting and character flaws identified and reported by OpenITI in the OCR output. These include:

- Formatting or font alterations related to text size (e.g., headers or footnotes), text placement (super- or sub-scripts), and bold or italicized text were the cause for a number of the output errors identified in the accuracy study.
- Certain characters such as the *Hamza*, punctuation marks, numbers, and non-alphanumeric symbols were a recurring source of errors.
- Atypical presentation or character patterns also resulted in output errors. One such example is the Arabic elongation character (*kashīda/tatwīl*).

OpenITI felt that such instances could best be addressed via more systematic training data creation that recognizes the presence of such characters and accounts for them in the text selected as the basis for training data. There were also technical issues described in OpenITI's report that could be addressed by improved line segmentation and layout analysis in the software. A recognition model that accommodated multiple languages/character systems was also identified for future investigation.

Addressing these technical challenges relates to the question of scale. Varying levels of effort and resources would be required to address them, and while it is not in JSTOR's purview to solve them, our participation in this project has provided one additional clear mechanism to identify such technical issues that would be worthy of investigation. Identification of such issues, using real-world examples and testing environments, such as the one created for this project, is a first step. As OpenITI considers how to scale its work, such projects will be valuable in surfacing aspects of Arabic digitization that are essential to solve.

Another challenge concerned the difficulty with establishing and adhering to a pre-determined timeline and the consequent unpredictability of turnaround times. Several factors contributed to this. First, this was a small investigative project, rather than a large-scale conversion project. The former implies unexpected obstacles, the identification of which is an important part of a successful investigation. In the latter, deadlines and predictability are crucial. Given the already full schedules for the OpenITI PIs and the OpenITI technical lead, it was sometimes difficult for them to respond to outstanding queries and tasks in a timely manner. When Apex was more directly involved, delays in responses from OpenITI meant that Apex would not know when to expect requested information, thereby creating inefficiencies. Predictability and time management are factors that must be addressed if Arabic journal digitization is to be operationalized at a larger scale. OpenITI suggests that project management for work they undertake could be resourced by JSTOR funding the time for a current (and appropriately skilled) staff member at one of their institutions or by funding the hiring of a new staff member devoted to project management. Likewise, OpenITI recommends broadening its technical team in order to provide support for the current lead who undertook the technical work for this project on his own. As with addressing some of the recurring flaws in the OCR output, there are costs associated with these approaches that are not part of expenses for JSTOR's standard conversion process, but they would be necessary to address in some way.

A different facet of the time-management challenge is the introduction of the essential yet time-consuming step(s) of reviewing the source pages for typefaces and creating corresponding training data for Kraken. This portion of the investigative project consumed roughly 3.5 months in order to create around 7,000 lines of training data. It is important not to use this as an indicator of exactly how long typeface identification and training data creation would require in a larger project. As OpenITI describes in its report, during the course of the 3.5 months they moved to a new and much more robust user interface for training data creation. This migration slowed down the work already underway.

OpenITI also notes that addressing potential gaps in the training data would require a more systematic approach to selection of text for training data creation. Doing so would ensure adequate coverage of text that, in the OCR output for this project, consistently generated output errors even after Kraken had been trained on the identified typefaces. A more systematic approach suggests increased effort (and time) would be needed, but the practical effect on turnaround times is not yet certain. Regardless, the results of this project indicate the clear value of the training data not only to Kraken's recognition of specific typefaces, but also to the generalized model, where output accuracy exceeds that of typeface-specific recognition models. Further, the implication is that, over time, as more typefaces are identified and training data is created, the generalized model should yield better and better accuracy such that it will be possible to create less training data for new typefaces, while still achieving high accuracy rates. Therefore, while initial investments in training data could be significant, expenses for this portion of the process should eventually decrease. This finding is a significant potential development in the cost-effectiveness of this approach over time.

Conclusions and Next Steps

Given these findings, what would a likely workflow for Arabic journal digitization in the context of JSTOR look like, how might it differ from JSTOR's standard journals workflow, and what are the implications for cost and overall practicality?

JSTOR calculates the per page cost for *al-Abhath*, excluding JSTOR staff time, to have been roughly $3. Activities informing this estimate are those required for operationalizing a larger scale project, such as: typeface identification, training data creation, creation and provision of OCR files, metadata capture, packaging of content for delivery, quality control of metadata at both Apex and JSTOR, and project management for the relevant activities at OpenITI and Apex. The portion of these costs attributable to Apex reflects both the long-standing working relationship that JSTOR and Apex have and any potential volume-based discounts that Apex extends to JSTOR.

This project included one-time activities that were specific to the collaboration described here. The estimated per page cost does not account for these one-time activities, such as the accuracy study conducted by OpenITI or the drafting of the Arabic metadata guidelines document by JSTOR.[14] As described above, some technical challenges may need to be investigated and solved before the workflow here can be fully operationalized at scale. This per page cost estimate does not account for any of these potential costs, such as workflow development by Apex or technical improvements to the Kraken software by OpenITI. Determining whether or not such costs are required and, if yes, how much they would be is a subsequent step.

Consequently, JSTOR's estimated per page cost may be most useful externally when put in the context of the range of tasks it does or doesn't cover, rather than as a firm guide for the costs for which others should budget when planning their own future projects. The one-time costs that other projects may incur cannot be generalized from the collaboration described here, but those costs were essential to the success of this project. Additionally, there is, broad benefit to the community at large from understanding what JSTOR's cost estimate is and, more importantly, what factors informed it. We hope that by transparently sharing all these activities in detail here, we may contribute to this understanding.

This cost is more than JSTOR's current standard per page costs. A large portion of the $3 per page cost is comprised of expenses for typeface recognition and training data creation. As stated above, if Kraken's generalized model is used, then eventually costs related to these activities should decrease. Given these higher costs, an area to examine is how the individual steps of the overall workflow could be adjusted holistically to account for these relatively higher costs. Presently, JSTOR moves *individual* journal back runs that are new to JSTOR through the licensing and conversion process. In an Arabic language context, a more compartmentalized approach could be taken wherein JSTOR would select a larger group of journals (e.g., anywhere from 10 to 15 journals to a

[14] It also does not account for costs related to acquiring print source issues or digital source files which would be required for a larger project. It is uncertain to what extent JSTOR could source digital files either from the publisher or from third parties (as we did in this instance, from Yale University/AMEEL, with permission from AUB) rather than scanning from print. If scanning from print, it is equally uncertain how much of a journal back run could be obtained by loans or donations from publisher or libraries vs. purchases from commercial vendors and antiquarian bookstores. Given that this cost is highly variable, further investigation will be needed to estimate source content acquisition costs.

much larger group), move all of them through each step of the licensing and conversion process, and then move all of them to the next step, until the group of journals has completed the full process and are available on JSTOR.

In this approach, the selection of journals to include is one that would need to be carefully done, using as broad a set of inputs from experts as possible. Our selection of *al-Abhath* was driven first and foremost by its high quality and its potential value to the scholarly community in the Middle East and globally. It was also driven by the institutional collaboration and relationship that was able to be established, and AUB's mutual interest in seeing Arabic journal content on JSTOR. It was not, however, driven by its potential for OCR accuracy improvements. It is possible that a further examination stage would be something to consider, to first seek texts that can be specifically useful in garnering further OCR accuracy improvements before selecting final journals to be included in any larger project.

In the context of a larger Arabic journals digitization project, there are multiple potential advantages to the latter, more holistic, approach:

- JSTOR expects the licensing process, which occurs at the beginning of the overall process, to be rather complex and lengthy. As many valuable existing projects are underway that focus on out-of-copyright materials, one of our aims is to seek journals that continue to publish. Waiting to move content into the subsequent steps of production will ensure that there is at least a minimal sufficient amount of content ready to be processed. If JSTOR put each journal into the conversion process as soon as a licensing agreement was concluded and source file or print volumes were available, then there could be gaps where no content was in the conversion pipeline, and this makes for an inefficient process overall. It should be noted, however, that this might require some management of publishers' expectations, and on thoughtful and considered relationship-building with those publishers who choose to participate.
- Review of multiple journal back runs might make it easier to identify use of similar or identical typefaces across titles, thereby reducing the amount of training data required.
- For OpenITI, Apex, and JSTOR, it can be challenging to devote time to sporadic and siloed workflows processing smaller amounts of content. While this may seem counterintuitive, the turnaround times in the current investigative project demonstrated it. Starting the production process with a larger group of journals doesn't mean that they all have to be processed at once. It does, however, provide a consistent flow of work that in turn can promote more effective time management.
- Given the success of Kraken's generalized recognition model, creating training data for a larger group of journals and *then* running OCR on the journals after Kraken has absorbed the training data would likely result in higher average OCR accuracy rates *across* that group of journals.

Should we be able to continue work with OpenITI and Apex, the good news is that costs related to training data creation could be expected to decrease as Kraken's generalized recognition model improves. Such improvements should then benefit the broader community of academic scholars of Arabic texts. Likewise, as noted above, the identification of key technical areas to be investigated for possible improvements is a vital first step.

In conclusion, JSTOR believes that the current project has indicated the potential for success in a larger digitization project. While new content and workflows will present unanticipated challenges, JSTOR's experience with large scale digitization as well as specialized projects leads us to believe that there are ways to foster greater consistency and predictability in workflows and high quality in the output. Our experience alone has not driven its success; our collaboration with OpenITI and the many institutions and individuals that contributed in myriad ways are essential. Indeed, this project emphasizes the need for strong collaboration that incorporates a broad range of entities from across the ecosystem of scholarly communications—from academic institutions and libraries, not-for-profit organizations, publishers, and even commercial enterprises. Our discussions regarding a possible future collaboration with OpenITI are ongoing. We hope that our participation in this project, and any future projects in this area, can be a contribution to the infrastructure of a community-driven, innovative endeavor to increase the amount of Arabic language scholarly content available online for research and teaching. While the findings here are specific to this project, we hope that the supporting methodology and practices will contribute to a growing collective knowledge.

As we seek to evaluate the potential for a larger project, we intend to examine a more holistic approach to the endeavor as a whole, as described above. While this investigation was limited to Arabic, the knowledge gained from this process has the potential to open a door to further areas of research toward other languages using this alphabet. Doing so is of interest to JSTOR, albeit on a much longer term, as it would begin to address the global representation to which we aspire, and ensure the long-term accessibility of these materials.

Appendices

Appendix A:

OpenITI's report, written by Matthew Miller to JSTOR regarding its work on typeface recognition, training data creation, OCR output, and related accuracy assessment thereof for selected issues of *al-Abhath*.

Open Islamicate Texts Initiative (OpenITI)

JSTOR OCR Pilot Report

I. Research Team Members[15]

OpenITI Principle Investigators

Matthew Thomas Miller (Roshan Institute for Persian Studies, University of Maryland)

Maxim Romanov (University of Vienna)

Sarah Bowen Savant (Aga Khan University, London)

OpenITI Technical Lead

Benjamin Kiessling (University of Leipzig/Université PSL)

OpenITI Research Assistants

Gennady Kurin (University of Maryland)

Majid Montazer Mahdi (University of Exeter)

Kader Smail (University of Maryland)

II. Background

[15] We would also like to thank David Smith (Northeastern University) for his suggestions and feedback on this report.

JSTOR contracted with OpenITI to run an OCR pilot for the JSTOR NEH Arabic digitization feasibility study. OpenITI is a multi-institutional initiative that is focused on building digital infrastructure for the computational study of the texts of the Islamicate world.[16] It is currently led by Dr. Matthew Thomas Miller (Roshan Institute for Persian Studies, University of Maryland, College Park), Dr. Sarah Bowen Savant (Aga Khan University, London), and Dr. Maxim Romanov (University of Vienna). Benjamin Kiessling (University of Leipzig/Université PSL) is one of OpenITI's primary computer science collaborators and he served as the technical lead for the OpenITI JSTOR OCR pilot.

III. OpenITI OCR Software

OpenITI uses an open-source OCR engine called Kraken that was developed by Kiessling. Kraken utilizes a neural network approach to perform OCR—specifically, the CLSTM neural network library—which obviates the need for character-level segmentation—i.e., the technical issue that has stymied the development of high-quality OCR for connected script languages more broadly. Rather than segmenting at the character level, Kraken operates at the level of the line, assigning labels (characters) to regions of the unsegmented data (lines). Using Kraken, OpenITI has achieved OCR results for several Arabic print typefaces in the high nineties with only a modest amount of training data.[17] This represents a substantial improvement over the actual OCR rates of the existing commercial solutions for Arabic-script OCR.

IV. The OpenITI JSTOR OCR Pilot

[16] More detail on the OpenITI project are available on the project's website: https://iti-corpus.github.io/.

[17] For full details on Kraken and the OCR results, please see: Benjamin Kiessling, Matthew Thomas Miller, Maxim Romanov, and Sarah Bowen Savant, "Important New Developments in Arabographic Optical Character Recognition (OCR)," *al-Usur al-Wusta* 25 (2017): 1-13, http://islamichistorycommons.org/mem/wp-content/uploads/sites/55/2017/11/UW-25-Savant-et-al.pdf (accessed 20 January 2019).

OpenITI began the JSTOR OCR pilot by performing a randomized review of the Arabic typefaces used in each year of the *al-Abhath* journal. It was determined that there were two basic typefaces in the *al-Abhath* journal archive, with the first typeface being much more prevalent than the second. Full results:

Typeface #1: volumes 1-33, 36-39, 48-50

Typeface #2: volumes 34-35, 40-47

Both typefaces had some internal font differences and other minor character/script variations (e.g., patterns of use of *alef hamza*, slight shifts in placement of dots, slight differences in degree of curvature of line in a couple of instances, and minor ligature differences). This intra-typeface variation was especially apparent in typeface #1, which had a long run as *al-Abhath's* typeface. To address this issue it was decided that the best approach would be to produce approximately 5,000 lines of training data for the first typeface and 2,000 lines of training data for the second typeface.

After a randomized sample of the pages representing each typeface were selected, the OpenITI team produced the training data for these 7,000 lines using CorpusBuilder (a new OCR postcorrection platform produced through the collaboration of OpenITI and Harvard's SHARIAsource project). After these 7,000 lines of training data were double checked for accuracy, the PIs conducted a final spot review. This "gold standard" data was then transferred to Kiessling for model production and OCR.[18]

Apart from OpenITI team's work, Apex CoVantage also helped in the preparation of the journal scans for the OpenITI pilot. Specifically, Apex:

(1) created the zone-level images that were provided to Open ITI as input for OCR;

(2) combined the OCR text data with the page images so that each page image (comprised of the various zones) has a corresponding searchable full text.

(3) provided JSTOR with the final issue-level package containing the page images, OCR files, and metadata. (It is worth noting that the work Apex performed in #1-3 may not actually be necessary in any future collaborations; it could, instead, (mostly) be incorporated

[18] This training data is available for reuse and can be found in the OCR Gold Standard Training Data repository on OpenITI's Github page: https://github.com/OpenITI/OCR_GS_Data.

into our workflows if we have sufficient time and resources to set it up.)

(4) conducted a ten-page manual accuracy comparison between the Kraken output and the corresponding output for ABBYY (see Table 1).

Table 1: Apex Accuracy Comparison of Abbyy and OpenITI (Kraken) OCR Results

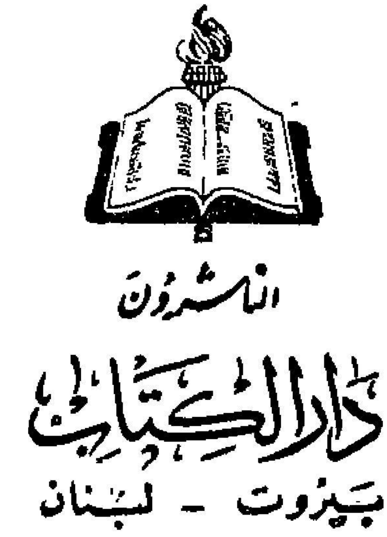
Page (Tiff) Number	Total Number of characters	Abbyy Character Errors	OpenITI Character Errors	Abbyy Accuracy Rate	OpenITI Accuracy Rate
Page #1 (00010004_187997831.tif)	1230	270	38	78.049%	96.911%
Page #2 (00010004_187997832.tif)	37	15	27	59.459%	27.027%
Page #3 (00010031_187998459.tif)	3182	355	23	88.843%	99.277%
Page #4 (00010063_187999338.tif)	3157	327	29	89.642%	99.081%
Page #5 (00010129_188001031.tif)	3222	378	16	88.268%	99.503%
Page #6 (00010012.tif)	3259	326	75	89.997%	97.699%
Page #7 (00010030.tif)	2503	230	17	90.811%	99.321%

Page #8 (00010126.tif)	2631	252	170	90.422%	93.539%
Page #9 (00010127.tif)	2294	223	35	90.279%	98.474%
Page #10 (00010132.tif)	2296	243	96	89.416%	95.819%

With the exception of page #2, OpenITI (Kraken) performed substantially better on the pages Apex reviewed, achieving >99% accuracy in 4/10 pages, >97% accuracy in 6/10 pages, >95.8% in 8/10 pages.[19] The exception to these generally impressive numbers were pages #2 and #8 in which OpenITI (Kraken) only achieved 27.027% and 93.539% respectively. While Apex's review was quite useful and generally confirmed OpenITI's results from its previous work (i.e., that Kraken achieves significantly higher accuracy rates on Arabic texts than the commercial OCR solutions for Arabic, see footnote #2), the OpenITI team discovered upon further review that there were several problems with Apex's study.

[19] The page numbers in the Apex accuracy study do not match the page numbers in the OpenITI manual accuracy, which are listed in Appendix I.

Figure#1: Page #2 Header



First, Page#2—by far the most disappointing result—is a highly atypical page of *al-Abhath* data. It only contains 37 characters total and much of these are contained in a large header that is in a highly calligraphic script and is heavily vocalized (with diacritics) (see figure #1). It is noteworthy that Abbyy performed better on this script, but this page is an extreme outlier in the data and Kraken's accuracy on this calligraphic script could easily be improved with the addition of further training data. The proposed improvements to the line segmenter in Kraken would also increase accuracy on such atypical headers as well.

The second issue that we identified in Apex's accuracy review was that they were marking certain differences between the original scans and OCR output as errors which were not true errors, and, in some cases, even marked some characters in the OCR output as errors that need not have been marked as errors at all. For example, in the former case, they marked all numbers as errors in the OpenITI OCR output which were rendered as western Arabic numerals (e.g., 1, 2, 3) instead of as eastern Arabic numerals (e.g., ١, ٢, ٣)—a problem that was particularly prevalent on page #8 (thus at least partially responsible for OpenITI's comparatively lower accuracy rate on page #8). The OCR rendered them as western Arabic numerals instead of eastern Arabic numerals because we decided to merge western and eastern Arabic numerals into their universal numerical values in the OCR process and then represent that value in western Arabic numerals in the OCR output.[20] These differences, thus, are not true errors—their numerical value is correct—and their representation can be changed to eastern Arabic numerals if that is what users prefer. Another similar issue was discovered in Apex's treatment of diacritics: they routinely marked correctly rendered words as incorrect if the word's original diacritics were not included in the OCR output. However, again, this difference in

[20] This practice of collapsing numeric values to their universal numerical value can be done for multiple reasons, but, in this particular case, one of our primary motivating factors was the fact that there were inconsistencies in the transcription practice of numbers in the training data.

the original text and the OCR output is not a true error in transcription because OpenITI has followed the practice (with one exception discussed further below) of not reproducing diacritics in its training data (for a number of reasons) and thus the fact that the diacritics were not rendered in OpenITI's OCR output is actually a sign that the Kraken OCR engine was functioning correctly. (This training data generation practice can be changed if the users desire, and given the results in OpenITI's larger accuracy study described below, this change may be advisable in the future, depending on the requirements of each individual user's use case.) Apex had not been informed that these differences between the original scan and the OCR output were expected and therefore did not constitute errors.

Apex's approach to error designation led them to calculate lower accuracy estimates for OpenITI OCR output than it achieved in actuality—a problem that was particularly accentuated in the case of page #8, which contained a larger amount of numbers than the other pages Apex reviewed. Once this was known and given OpenITI's familiarity with its OCR output expectations, JSTOR requested that OpenITI perform a more detailed accuracy assessment on approximately one hundred pages of the OCR output. In the course of this study it was mutually agreed upon that the number of pages to be reviewed would be reduced to fifty pages due to time constraints.

V. OpenITI Accuracy Study

The OpenITI team began by generating automatic character error rate (CER) reports for the *al-Abhath* data (see Table 2 for full results).[21] In the first round of experiments, Kiessling built two different models—typeface model #1 and #2—based on the two different sets of training data produced for the two typefaces that we identified in the full run of *al-Abhath*. After extracting 1,000 lines of training data from the 5,000 lines of training data for typeface #1 and 700 lines of training data from the 2,000 lines of training data for typeface #2 to use as validation sets, he then trained the model on the remaining lines and tested these models' accuracy using the validation sets.[22] These accuracy results can be found

[21] The full CER reports can be found in the following OpenITI Github repository: https://github.com/OpenITI/OCR_GS_Data/tree/master/ara/abhath.

[22] This method of isolating a fixed number of lines of the training data as a validation set for automatic accuracy testing is a standard procedure when evaluating machine learning models.

in rows 2-3 in Table 2. These typeface-specific models were the ones used to produce the OCR output that was transferred to JSTOR and that Apex reviewed for their accuracy study.

In the time between the delivery of the *al-Abhath* OCR output to JSTOR and OpenITI's manual accuracy study (discussed below), Kiessling began developing a generalized Arabic model from all of the training data that OpenITI has produced over the last year two years (circa 15,000 lines).[23] (Generalized OCR models incorporate character features from all of the typefaces represented in the data upon which it is trained and therefore can often achieve higher levels of accuracy on a broader range of typefaces.) We decided to test this model on all of the *al-Abhath* data to determine if total OCR accuracy could be improved and, if so, by how much. The results, shown in row #4 of Table 2, were impressive.[24] The generalized model's total character accuracy rate was 97.41%—a 2.57% improvement over the typeface #2-only model (i.e., a ~50% improvement rate) and 1.45% improvement over the typeface #1-only model—and its Arabic script-only accuracy went up to a respectable 98.46%. The generalized model performed better than the typeface-specific models in all categories, but its most significant gains were in the category of "inherited" characters.

[23] All of this gold standard training data can be found here: https://github.com/OpenITI/OCR_GS_Data.

[24] For this accuracy assessment, 2,096 lines of the 7,000 lines of training data were isolated as a validation set.

Table 2: Overview of OCR Accuracy Rates (Drawn from Character Error Rate (CER) Reports)

Model	**Total Character Accuracy**	**Arabic Script Only Accuracy**	**Common Character Accuracy***	**Inherited Character Accuracy****
Typeface #1 Model	95.96%	97.56%	96.91%	79.67%
Typeface #2 Model	94.84%	97.11%	94.16%	85.18%
Generalized Model	97.41%	98.46%	96.36%	89.44%

** “Common characters” are characters shared by multiple scripts, primarily punctuation and other signs and symbols. In Arabic script, the* kashīda *or* tatwīl *(elongation character) is included in common script class.*

*** “Inherited characters” are characters, such as diacritics, that can be used on multiple languages and they only come to be defined in reference to the character with which they are combined (i.e., they “inherit” the script of the base character with which it is used).*

According to the CER reports, the most significant source of errors in both the typeface #2 and generalized models were whitespace (spacing) errors and the Arabic diacritic, *fatḥa tanwīn* (unicode codepoint: Arabic *fatḥatan*). In the case of the typeface #1 model, whitespace errors were again the most significant source of errors, followed by *kāf* (ك), *yā’* (ي), and then *fatḥa tanwīn* errors. The *hamza above* (ٔ) character ranks as the seventh most common error in typeface #1 model and fifth in typeface #2 model. The *mīm* (م) character also is a common error in both the typeface #1 and #2 models, ranking as the sixth and fifth most common error in their CER reports respectively.[25]

[25] Again, the full CER reports can be found in the following OpenITI Github repository: https://github.com/OpenITI/OCR_GS_Data/tree/master/ara/abhath.

Concurrent with Kiessling's generation of CER reports, the other members of the OpenITI team began a far more expansive manual review of fifty—randomly selected—pages of the original OCR output produced by the typeface #1 and typeface #2-specific models. Two separate individuals reviewed each of these fifty pages and then their error reports were collated by a third individual into a master list of 1,096 total error instances.[26] Finally, a fourth individual re-examined each error instance with an eye towards identifying possible factors in its adjacent context that may have led to that error and coded the error instances with any of the following categories that were applicable:

1) ***Poor scan quality:*** an element in the raw scan is unclear, or extraneous marks are present.
2) ***Ligature/atypical letter or dot form:*** connection between letters or placement of dots is in a less common form.
3) ***Diacritics:*** diacritics were present in original word.
4) **Kashīda/tatwīl *(elongation character):*** error appears in the context of a word that has been elongated.
5) ***Header/font Alteration:*** bolded, italicized, or enlarged text.
6) ***Footnote:*** error appears in the context of a footnote.
7) ***Format:*** atypical format of presentation, e.g., table, list.
8) ***Hamza:*** mistranscribed character was a *hamza* or a *hamza* was present in original word that was mistranscribed.
9) ***Doubled character:*** a single letter or number in the original scan was doubled in the OCR output.
10) ***Missed* fatḥa tanwīn*:*** *fatḥa tanwīn* in the original text was not transcribed.
11) ***Punctuation or other symbol:*** error was a punctuation mark or other symbol.
12) ***Non-Arabic language:*** original text was not Arabic.
13) ***Numbers:*** error was a number.
14) ***Superscript numerals:*** error was a footnote numeral in the body of the text.

[26] We use the term "error instance" here to highlight the fact that we are not exclusively recording individual, one-to-one character errors, but instances in the text in which one or more characters were read incorrectly. In most cases, this is a one-to-one character mistranscription, but in some other cases in is one character in the original read as two or more in the OCR output or multiple characters in the original that are read as one or none in the OCR output. In a few cases—discussed in more detail below—there are whole sections of text that are severely mistranscribed due to one or another feature in the original text.

This list of error codes is a mixture of error types (#8-14) and the most common recurring contextual features of the errors (#1-7). For categories of the latter type, it is important to emphasize that the presence of any of these contextual features near an error in the original text does not necessarily mean that it *caused* the error. But their repeated co-occurrence may be related and thus suggest future avenues of research and/or the need to better address this issue in the process of future training data production. We should also point out that in the case of some errors none of the following category codes were applicable, which only means that the reason for their improper rendering was not immediately evident to the human reviewers.

We do want to preface our presentation of the results of this manual review and error coding below with one further cautionary note. Manual evaluations are both essential and problematic: they provide far more detailed data (i.e., “thick data”) about the OCR output and where OCR is failing, but they are much more time and resource intensive (and thus more limited in scope) and subject to human error. The results presented in Table 3 should be understood in this light. They should be understood as a snapshot of the human-inferable errors present in the OCR output. Each error type and possible ways to address it will be discussed in more detail in separate sections below Table 3.

Table 3: Error Coding for Error Instances in OpenITI Manual OCR Output Assessment

Error Code	**Quantity Identified**
Poor scan quality	25
Ligature/Atypical letter or dot form	182
Diacritics	90
Kashīda/tatwīl (elongation character)	31
Header/Font alteration	113

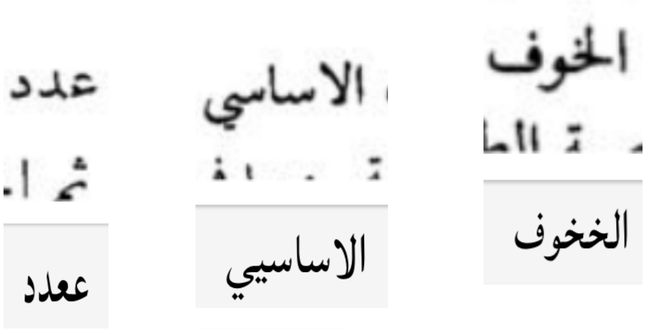
Footnote	88
Format	14
Hamza	97
Doubled letter	209
Missed *fatḥa tanwīn*	91
Punctuation or other non-alphanumeric symbols	25
Non-Arabic language	70
Numbers	94
Superscript numerals	26

Doubled Letter

The “doubled letter” error type was the most frequent that we observed in the OCR output data (see example in figure 2).[27]

[27] In the images in figures 2-29 below the Arabic text at the top of the images is the original scan and the text below is the OCR output. Also, the corresponding original file names for the “page number” listed for each image in the figure title can be found in Appendix I.

Figure 2: “Doubled Letter” Errors (pages 98, 97, and 97, respectively)

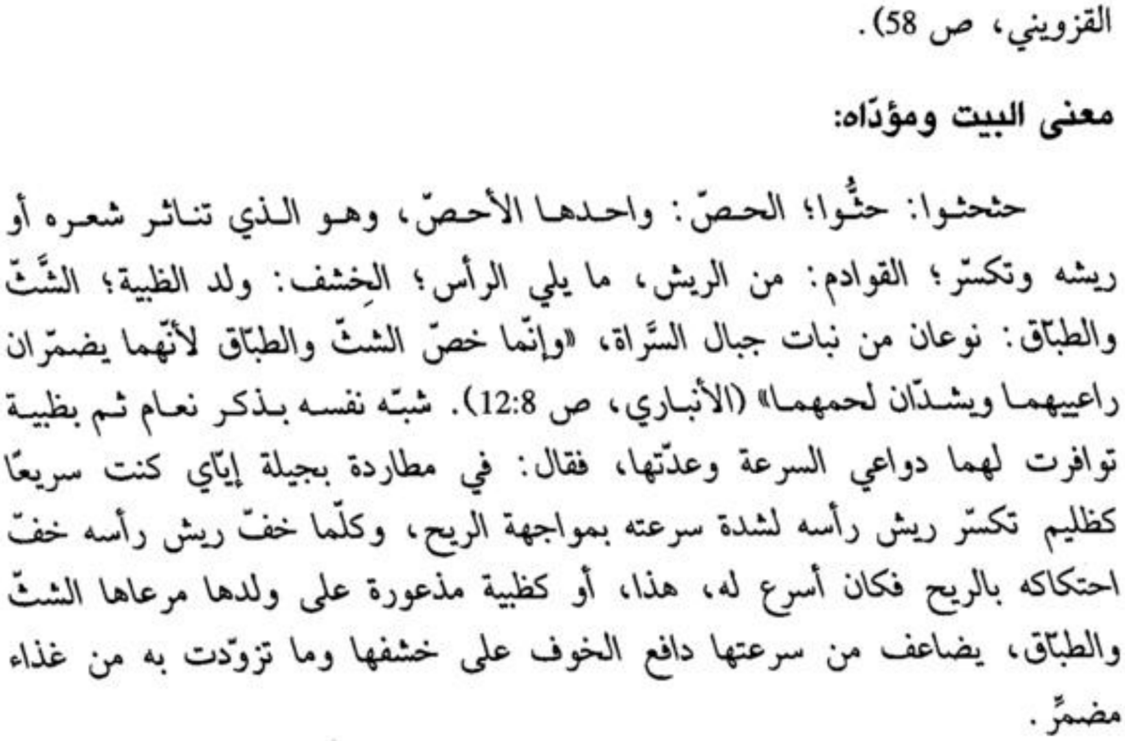


At first this error was perplexing. However, it was subsequently discovered that these “doubling” errors were an artifact caused by the particular shape of the neural network’s activations induced by the loss function during training and the thresholded character decoding algorithm converting the probability distribution in time from the network into a series of characters. Errors of this type will not be a problem in future work.

Header/Font Alteration, Footnotes, and Superscript Numerals

Errors that occured in the context of changes in the font (bolded, italicized, enlarged/decreased text size) represent the largest category of errors in the OCR output. Their total numbers are not even fully reflected in Table 3 because examples in which whole sections (see examples in figures 3-4) were severely mistranscribed were not enumerated (character-by-character) in the error totals of the OpenITI manual assessment. Such sections were very rare, and in most other cases the OCR still rendered text with font alterations with a relatively high degree of accuracy, but font alterations do seem to increase error rates.

Figure 3: Example of text in italics (Page 91)

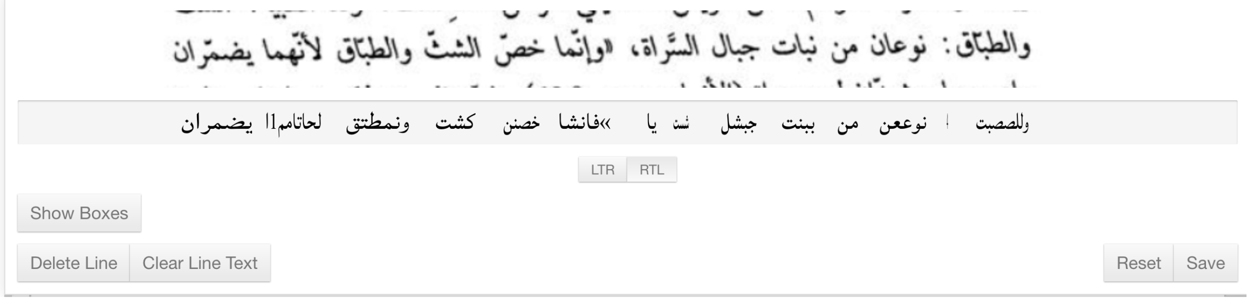


القزويني، ص 58).

معنى البيت ومؤدّاه:

حنحشوا: حثُّوا؛ الحصّ: واحدها الأحصّ، وهو الذي تناثر شعره أو
ريشه وتكسّر؛ القوادم: من الريش، ما يلي الرأس؛ الخِشف: ولد الظبية؛ الشَّثّ
والطبّاق: نوعان من نبات جبال السَّراة، «وإنّما خصّ الشثّ والطبّاق لأنّهما يضمّران
راعييهما ويشدّان لحمهما» (الأنباري، ص 12:8). شبّه نفسه بذكر نعام ثم بظبية
توافرت لهما دواعي السرعة وعدّتها، فقال: في مطاردة بجيلة إيّاي كنت سريعًا
كظليم تكسّر ريش رأسه لشدة سرعته بمواجهة الريح، وكلّما خفّ ريش رأسه خفّ
احتكاكه بالريح فكان أسرع له، هذا، أو كظبية مذعورة على ولدها مرعاها الشثّ
والطبّاق، يضاعف من سرعتها دافع الخوف على خشفها وما تزوّدت به من غذاء
مضمّرٍ.

Figure 4: Example of poor transcription of italicized passage in figure 3.

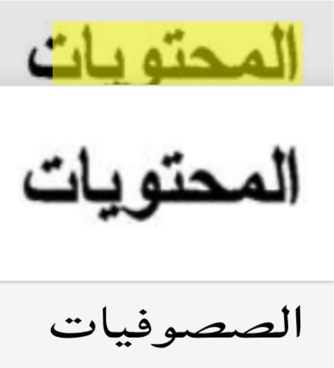


والطبّاق: نوعان من نبات جبال السَّراة، «وإنّما خصّ الشثّ والطبّاق لأنّهما يضمّران

وللصبت ! نوععن من ببنت جبشل ئسن يا »فانشا خصنن كشت ونمطتق لحاتامI1 يضمران

One of the most common examples of this issue was observed in text headers (including both section headers and chapter titles), which were typically bolded or bolded and enlarged in *al-Abhath* (see figures 5-6). In some headers, as mentioned above, an entirely different typeface was employed (see figure 1)—although this is a less common practice.

Figure 5: Bolded and enlarged text size header and poor transcription (page 99):

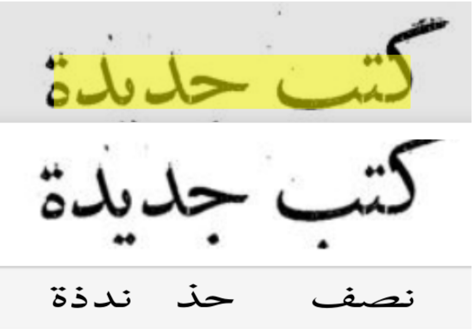


المحتويات

المحتويات

الصصوفيات

Figure 6: Bolded and enlarged text size header and poor transcription (page 70):



كتب حديدة

كتب جديدة

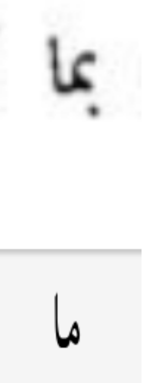
نصف حذ ندذة

Other modifications to the font of the typeface, e.g., footnotes, superscript (decreased text size) numerals, also seem to be correlated with decreased accuracy rates. In the future, this could be addressed by ensuring that a sufficient number of lines of training data with such font modifications is included.

Ligatures/Atypical Letter or Dot Forms

Not surprisingly, ligatures and other types of less common letter patterns and dot placements led to less accurate transcriptions in the OCR output (see figures 7-12). We observed this same problem in our 2017 study as well (see footnote 3).

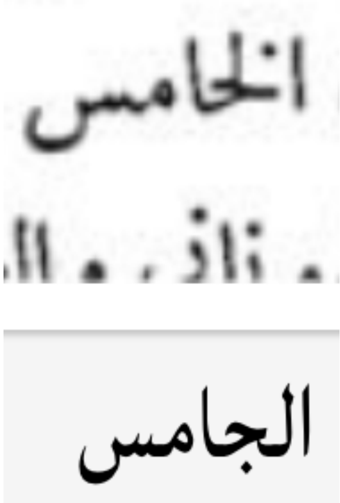
Figure 7: Example of problematic ligature and error in transcription (page 104)



بما

ما

Figure 8: Example of problematic ligature and error in transcription (page 79)



الخامس

الجامس

Figure 9: Atypical dot placement (page 72)

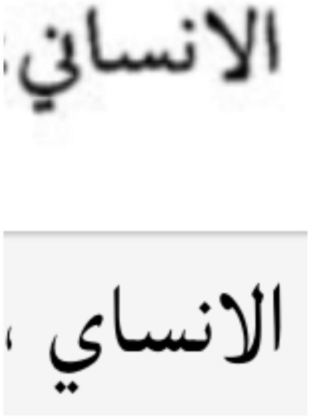

الانساني
الانساي

Figure 10: Atypical dot placement (page 79)

Figure 11: Atypical letter pattern (printing error?) (page 68)

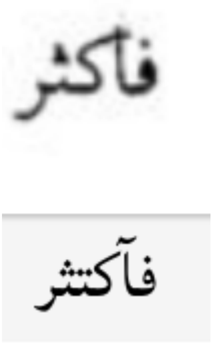
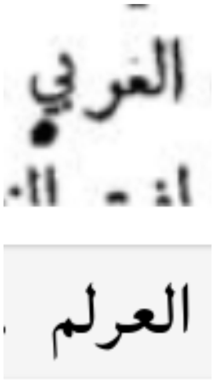

فاكثر
فآكتثر

Figure 12: Atypical dot/letter placement and poor scan quality (page 70)

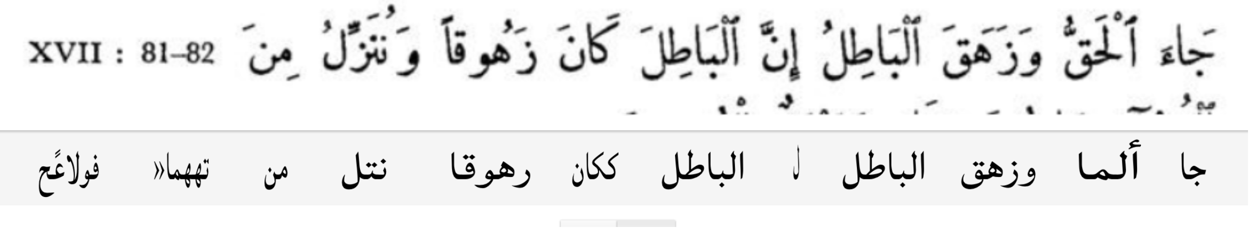


It is nearly impossible to completely avoid this problem, but a more systematic approach to training data generation that selected pages/lines of data with an eye towards ensuring sufficient representation of the maximum number of ligatures could improve OCR accuracy on these characters/character combinations.

Diacritics

Words that contained diacritics also appear more frequently to have errors in transcription, which leads us to believe that diacritics are interfering with character recognition. This tendency especially can be seen in examples of heavy vocalization, such as the fully vocalized Qur'anic passage seen in figure 13, which are poorly transcribed. Figure 13 is an extreme case that is an outlier in the *al-Abhath* data, but it clearly illustrates this problem. Moreover, although *al-Abhath* journal articles are not heavily vocalized, this could be a significant issue in other Arabic texts that are heavily vocalized.

Figure 13: Highly vocalized Qur'anic passage (page 86) that is transcribed poorly due to diacritics



جَاءَ ٱلْحَقُّ وَزَهَقَ ٱلْبَاطِلُ إِنَّ ٱلْبَاطِلَ كَانَ زَهُوقاً وَنُنَزِّلُ مِنَ XVII : 81–82

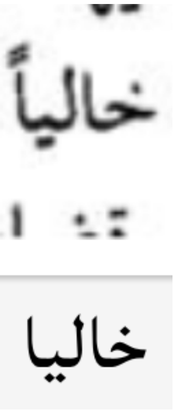
جا ألما وزهق الباطل ا الباطل ككان رهوقا نتل من تههما« فولاعًح

In general, OpenITI has traditionally followed the practice of *not* transcribing Arabic diacritics in our training data production (with one exception discussed below). We have followed this practice for three reasons: (1) vocalization is often inconsistent and sometimes incorrect (so it is better to allow the individual scholar to determine the proper vocalization based upon their reading); (2) vocalization can interfere with computational textual analysis (computational linguists, for example, typically remove it); and, (3) not all full-text search algorithms support diacritics in a useful way. There is one problem with this approach, however, that we have found in both this study and another concurrent one on Persian OCR. If there is a sufficient amount of diacritics in the original text, the model will "learn" to ignore diacritical marks and it will not interfere with character recognition. However, if the original text is lightly vocalized and not enough examples of diacritics are contained in the training data, then it appears that the model does not "learn" well enough to ignore the diacritics and thus their presence in a word interferes with accurate character recognition. This situation presents us with a dilemma around which we need to develop a set of guidelines: we do not want to include diacritics because of the aforementioned reasons and because including them in the training data will require even more time expenditure in the training data generation process, but by *not* including them in texts without heavy vocalization (e.g., *al-abhath*, some of the Persian texts in our other study) character recognition is reduced in words with them.

Missed *Fatḥa Tanwīn*

The exception to our traditional treatment of diacritics discussed in the previous section is the case of the Arabic diacritic *fatḥa tanwīn* (اً). As observed in the CER reports, missed *fatḥa tanwīns* were a significant source of errors. We also observed this in the manual review of the OCR output (see figure 14).

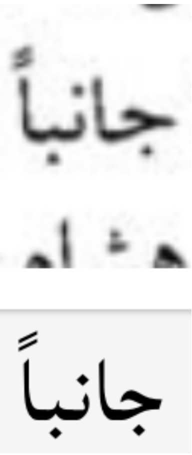
Figure 14: Missed* fatḥa tanwīn *(page 101)



خاليا

Although in the past we have not transcribed *fatḥa tanwīns* in the training data production process, we did include *fatḥa tanwīns* in the JSTOR pilot training data. In many cases the *fatḥa tanwīns* were transcribed correctly (see figure 15). However, as both the CER reports and manual review showed, they still remained a relatively common source of errors. The reason(s) that *fatḥa tanwīn* remained a problem in the transcription process could be related to either (1) its lack of sufficient representation in the training data, or (2) its position in the line segment—i.e., it might be partially getting cut off since it appears so high in the line segment box. In either case, we are inclined to ignore *fatḥa tanwīns* in future training data production.

Figure 15: Correctly transcribed* fatḥa tanwīn *from same page as figure 14 (page 101)



جانباً

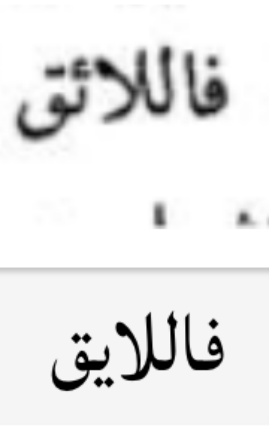
Punctuation Marks, Number, and Other Non-Alphanumeric Symbols

Punctuation marks, numbers, and other non-alphanumeric symbols (e.g., $)—especially representatives of each of these categories that were less commonly used in *al-Abhath*—were another recurring source of errors. The way to address this problem is by making sure these signs, symbols, and numbers are sufficiently represented in the training data.

Hamzas

The *hamza* character was another common source of errors in the output, both in the sense that it was misrecognized (see figure 16-17) and inserted in instances in which it was not in the original scan (see figure 18).

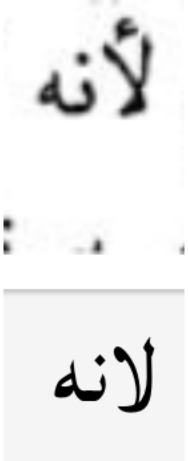
Figure 16: Missed* hamza *(page 78)



فاللائق

فاللايق

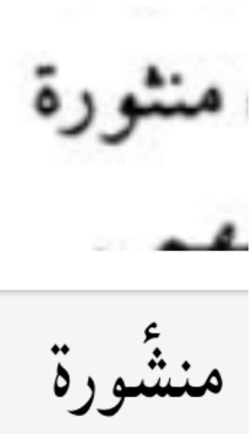
Figure 17: Missed **hamza** *on alif (page 98)*



لأنه

لانه

Figure 18: Inserted extra **hamza** *(page 101)*



منشورة

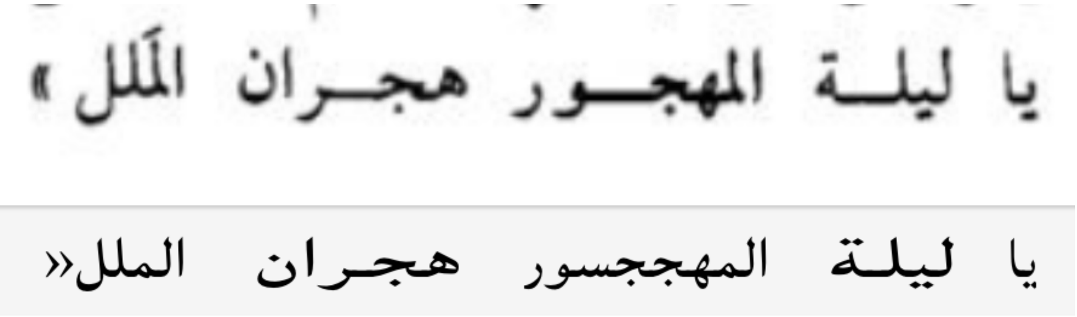
منشٔورة

Again, this is a case in which more training data will improve recognition rates—an intervention we must make at the training data generation phase of the OCR process.

Atypical Text Presentation Format and *Kashīda/tatwīl* (elongation character)

There are a series of errors that occur in the context of atypical presentation formats/atypical character patterns. These range from the use of the Arabic elongation character (*kashīda/tatwīl*) (see figure 19) to various types of table formats (see figures 20-22).

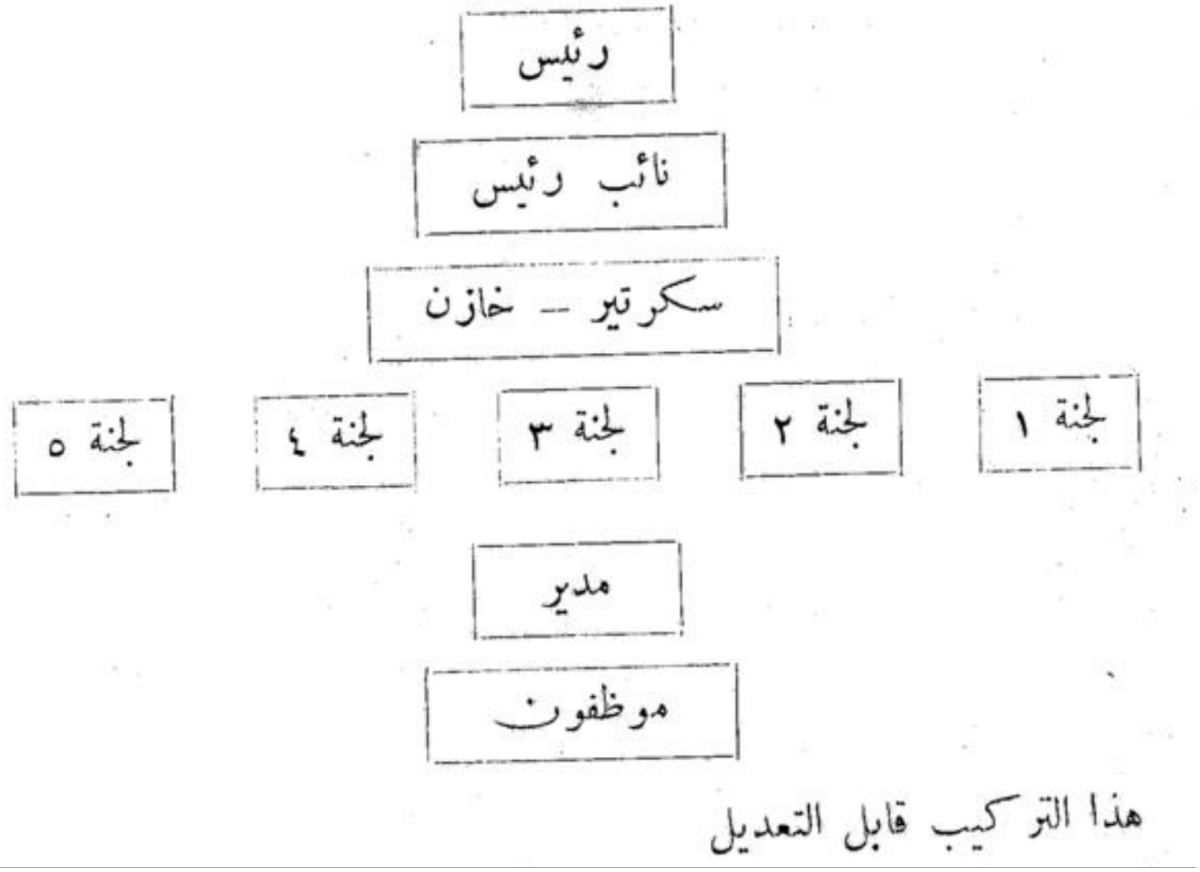
Figure 19: Read letter 'sin' into word due to **kashīda/tatwīl** *(elongation) (page 80)*



يا ليلــة المهجــور هجــران المَلل »

يا ليلـة المهجسور هجـران الملل«

Figure 20: Example of table format (page 87)



رئيس

نائب رئيس

سكرتير – خازن

لجنة ١

لجنة ٢

لجنة ٣

لجنة ٤

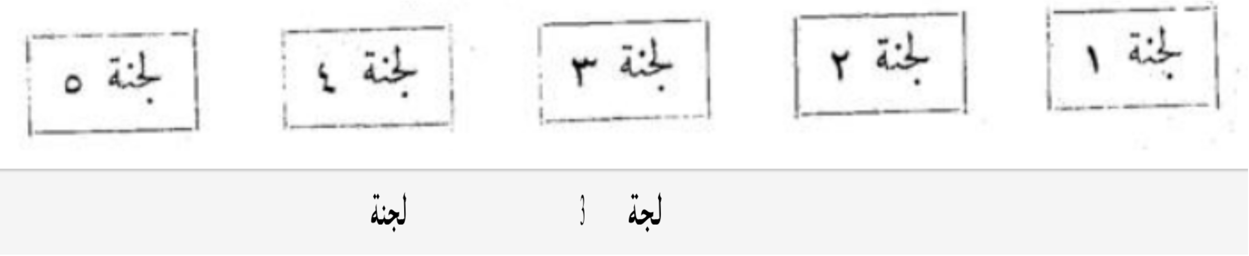
لجنة ٥

مدير

موظفون

هذا التركيب قابل التعديل

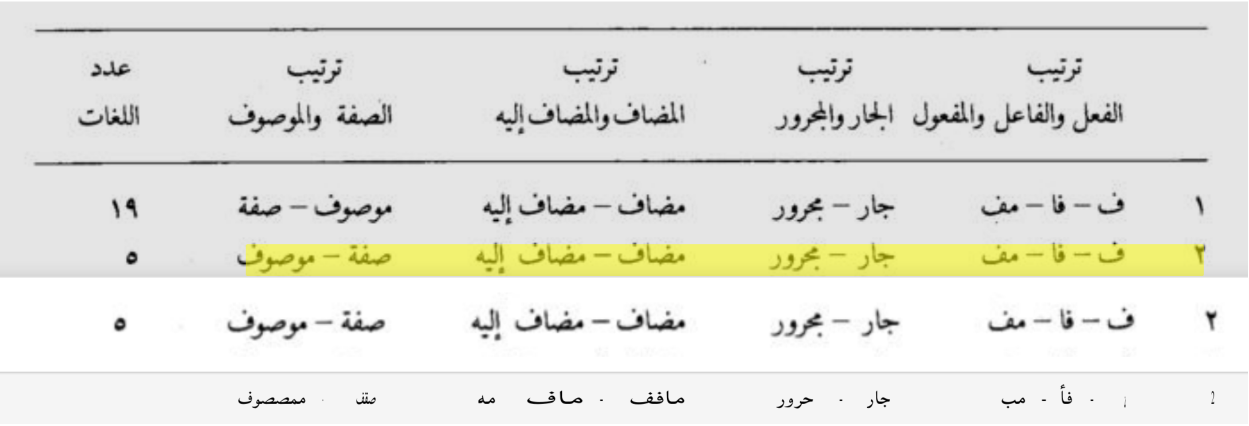
Figure 21: Example of particularly poor transcription on an atypical (table) presentation format (page 87)



لجنة ١	لجنة ٢	لجنة ٣	لجنة ٤	لجنة ٥

لجة) لجنة

Figure 22: Example of particularly poor transcription on another atypical (table) presentation format (page 105)



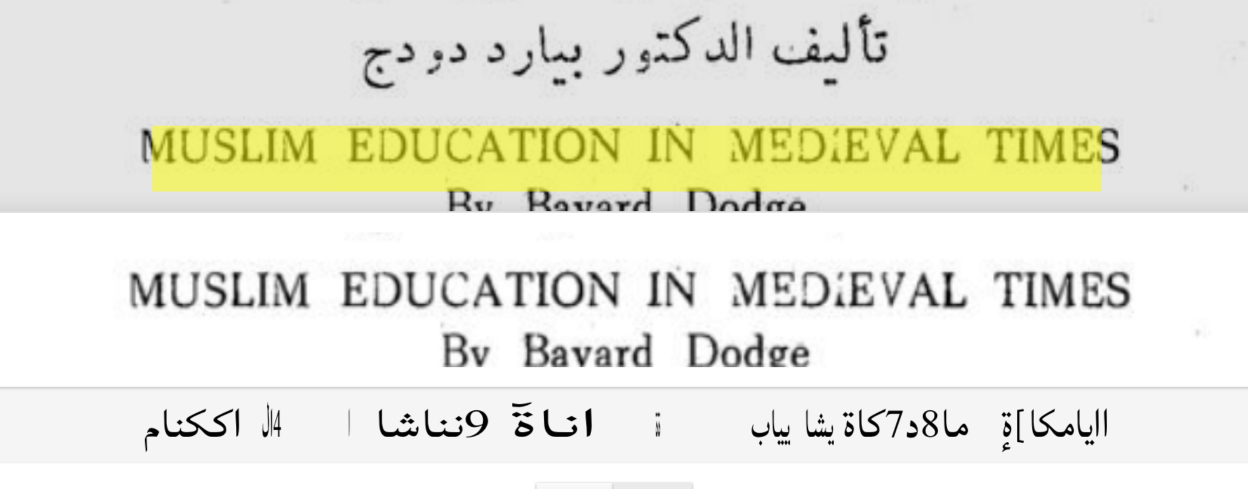
	ترتيب الفعل والفاعل والمفعول	ترتيب الجار والمجرور	ترتيب المضاف والمضاف إليه	ترتيب الصفة والموصوف	عدد اللغات
١	ف – فا – مف	جار – مجرور	مضاف – مضاف إليه	موصوف – صفة	١٩
٢	ف – فا – مف	جار – مجرور	مضاف – مضاف إليه	صفة – موصوف	٥
٢	ف – فا – مف	جار – مجرور	مضاف – مضاف إليه	صفة – موصوف	٥
2	ا - فأ - مب	جار - حرور	مافف - صاف مه	صقق · ممصصوف	

Although the character recognition in these examples is usually not as poor as in figure 21, we still observed that errors seem to appear more frequently in such contexts (see the better recognition in figures 19 and 22). More training data from these atypical presentation formats and character patterns will help improve accuracy, but improvements in line segmentation are also necessary for such examples as figure 21.

Non-Arabic Language

There were two significant types of transcription errors that were related to the presence of non-Arabic language in the original text. The first, seen in figure 23, is the poor transcription of non-Arabic characters on a page that predominantly contains Arabic text. (Figure 23 represents a particularly poor transcription of the non-Arabic text; most transcriptions in such instances were much more accurate.)

Figure 23: Example of particularly poor transcription of non-Arabic language in a page of primarily Arabic text (page 107)



تأليف الدكتور بيارد دودج

MUSLIM EDUCATION IN MEDiEVAL TIMES

Bv Bavard Dodge

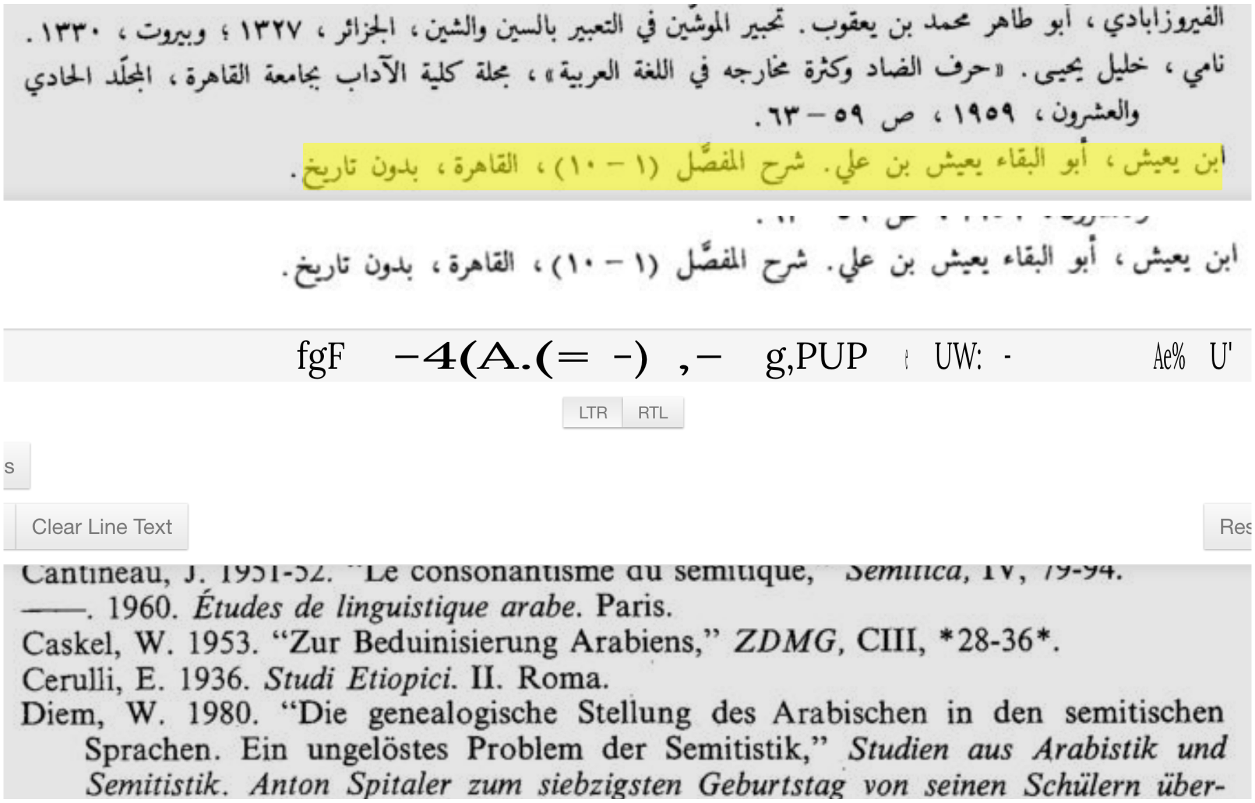
MUSLIM EDUCATION IN MEDiEVAL TIMES

Bv Bavard Dodge

اايامكا]ۃ ما8د7كاة يشا يياب انـاةَ 9نناشا | ال4 اككنام

The second type of error that occured in the context of non-Arabic script was the inverse: that is, poor transcription of Arabic text on a page that is predominantly composed of a non-Arabic language (see figure 24).

Figure 24: Page with substantial Non-Arabic language interferes with Arabic OCR (page 82)



الفيروزابادي ، أبو طاهر محمد بن يعقوب. تحبير الموشّين في التعبير بالسين والشين ، الجزائر ، ١٣٢٧ ؛ وبيروت ، ١٣٣٠.
نامي ، خليل يحيى. «حرف الضاد وكثرة مخارجه في اللغة العربية» ، مجلة كلية الآداب بجامعة القاهرة ، المجلّد الحادي والعشرون ، ١٩٥٩ ، ص ٥٩ – ٦٣.
ابن يعيش ، أبو البقاء يعيش بن علي. شرح المفصّل (١ – ١٠) ، القاهرة ، بدون تاريخ.

ابن يعيش ، أبو البقاء يعيش بن علي. شرح المفصّل (١ – ١٠) ، القاهرة ، بدون تاريخ.

fgF −4(A.(= -) ,− g,PUP ŧ UW: - Ae% U'

Cantineau, J. 1951-52. "Le consonantisme du sémitique," *Semitica*, IV, 79-94.
——. 1960. *Études de linguistique arabe*. Paris.
Caskel, W. 1953. "Zur Beduinisierung Arabiens," *ZDMG*, CIII, *28-36*.
Cerulli, E. 1936. *Studi Etiopici*. II. Roma.
Diem, W. 1980. "Die genealogische Stellung des Arabischen in den semitischen Sprachen. Ein ungelöstes Problem der Semitistik," *Studien aus Arabistik und Semitistik. Anton Spitaler zum siebzigsten Geburtstag von seinen Schülern über-*

This is a known problem that can be addressed through the development of multi-language OCR models—a project that we plan to undertake in the future as time and resources permit.

Poor Scan Quality

Poor scan/print quality—including, errant marks (see figure 25), lack of ink (see figure 26), misplaced letters/punctuation (figure 11)—is not a particularly common source of errors in the *al-Abhath* data, but there is a critical mass of errors caused by this problem.

Figure 25: Example of poor scan quality—black shading in background of letters (page 106)

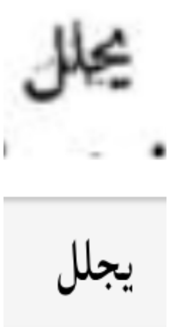
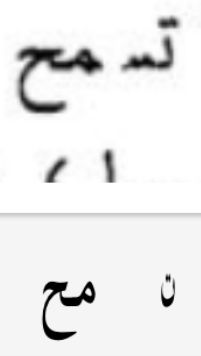


Figure 26: Example of poor scan quality—missing print in letter (page 98)

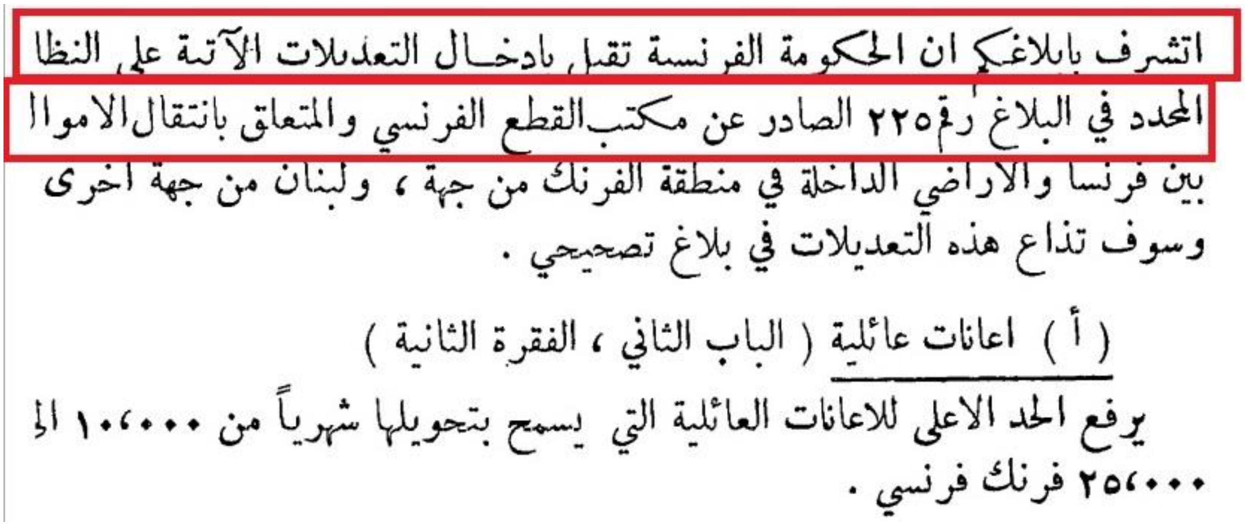


This problem cannot be addressed in the OCR process. OCR accuracy is (obviously) limited by the quality of the original scans.

Line Segmentation

One final error type that should be mentioned is line segmentation errors (see figures 27-29).

Figure 27: Missed line segments (from Apex accuracy review)

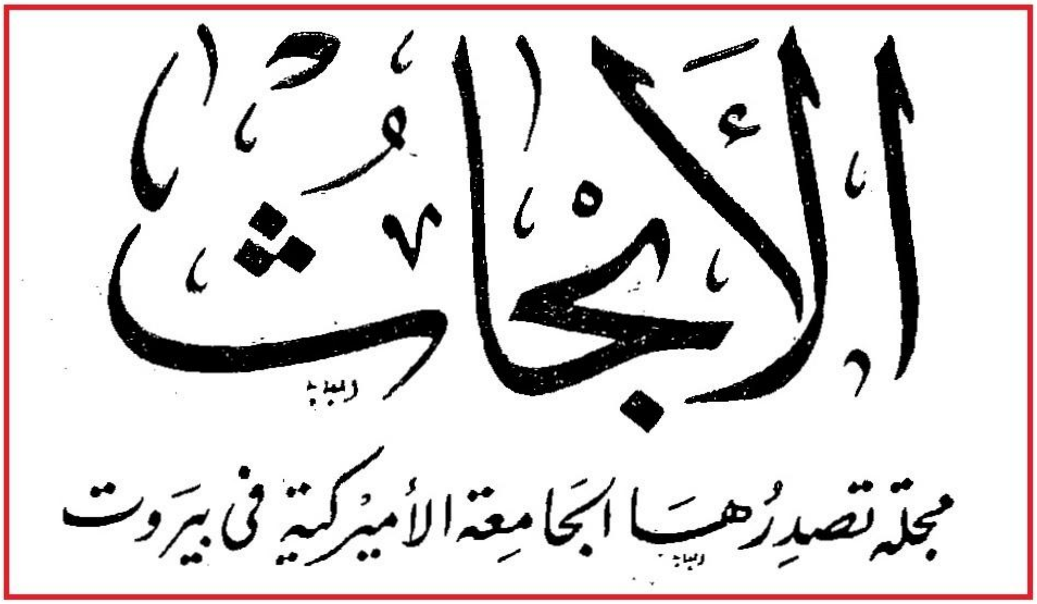


اتشرف بابلاغكم ان الحكومة الفرنسية تقبل بادخـال التعديلات الآتية على النظا
المحدد في البلاغ رقم ٢٢٥ الصادر عن مكتب القطع الفرنسي والمتعلق بانتقال الاموا
بين فرنسا والاراضي الداخلة في منطقة الفرنك من جهة ، ولبنان من جهة اخرى
وسوف تذاع هذه التعديلات في بلاغ تصحيحي .

(أ) اعانات عائلية (الباب الثاني ، الفقرة الثانية)

يرفع الحد الاعلى للاعانات العائلية التي يسمح بتحويلها شهرياً من ١٠،٠٠٠ الـ
٢٥،٠٠٠ فرنك فرنسي .

Figure 28: Large header segmented as one line (from Apex accuracy review)

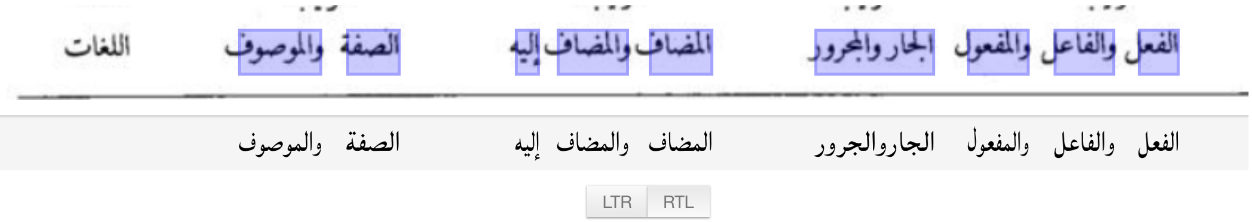


الأبحاث

مجلة تصدرها الجامعة الأميركية في بيروت

This type of error was not commonly found in the OpenITI manual accuracy assessment (figures 27-28 were errors identified in Apex's review of the OCR output), but there were a few cases in which the line segmenter missed a section or a word of a line. Typically this would occur in atypical text presentation formats, such as the table seen in figure 29.

Figure 29: Line segmenter missed final word in the line (page)


الفعل والفاعل والمفعول الجار والمجرور المضاف والمضاف إليه الصفة والموصوف اللغات
LTR
RTL

الفعل والفاعل والمفعول الجاروالجرور المضاف والمضاف إليه الصفة والموصوف

This is a known problem for the line segmenter in the version of Kraken that was used for this study. The solution for this problem is known and can be implemented with a modest amount of time dedicated to development.

VI. Recommendations and Future Avenues

In reviewing the results of the *al-Abhath* OCR output, there are three areas in which we believe we need to focus in future:

(1) **Systematic training data production.** Instead of generating training data in a completely randomized (or haphazard) manner (as we have done in the past), we need to study the particularities of the documents we plan to OCR and make sure that the pages selected for training data production contain a sufficient number of the less common ligatures, headers, diacritics, footnote text, numbers, and other particularities of the works to be OCRed. This more systematic approach to training data production will require more time upfront. But the models produced in this manner could potentially achieve much higher baseline accuracy and reduce the burden of postcorrection. Similarly, we need to develop a strict set of guidelines for training data producers regarding the transcription of diacritics, numbers, and non-alphanumeric symbols.

(2) **Generalized models can significantly improve accuracy.** One of the most exciting results from this study was the significant improvements in accuracy we achieved with the generalized model. The success of this approach tentatively suggests that if we continue to add training data sets to this generalized model we can anticipate to achieve higher levels of accuracy on *both* typefaces on which we have already trained models *and* new typefaces for which we have no training data yet. If this pattern holds true in future studies, we would be able to gradually reduce the time and resources necessary to achieve

high level accuracy (>97%) on new typefaces in the future. (However, more research on generalized models is needed.)

(3) **There are a range of technical improvements—e.g., multi-language models, improved line segmentation and layout analysis—that could significantly improve OCR accuracy numbers.**

VII. Challenges

The OpenITI OCR pilot was broadly successful. However, the pilot gave us the opportunity to work through a number of difficulties that need to be addressed in any possible future collaborations:

(1) *Project management:* The burden of project management fell primarily on the UMD PI, Matthew Miller. Given the experimental nature of this pilot, Dr. Miller elected to take on this responsibility in order to both develop a deeper sense of the workflow and ensure successful completion. However, in any future collaborations—especially if of a larger scale—the project management duties would need to be handled by a professional project manager through either a time buyout of existing staff time or a new hire (depending on the size of the project).

(2) *Diversification of technical team:* The technical lead in this pilot, Benjamin Kiessling, is both the developer of the OCR software used in this pilot and a leading figure in the field of OCR for non-Latin script languages. These facts make him an obvious choice for this position. However, for the same reasons, he is very busy and currently has a full-time position with the research team of Professor Daniel Stoekl of the Ecole Pratique des Hautes Etudes. In any future collaborations it would be ideal to have another member of our technical team who could assist Kiessling with some of the technical work so that the entire burden of the technical work does not fall upon him alone.

(3) *Technical issues:* The commencement of this pilot coincided with OpenITI's transition from a Pybossa-based OCR post-correction interface to a new open-source interface called CorpusBuilder. CorpusBuilder—the result of an innovative collaboration between OpenITI and Harvard's SHARIAsource project—is a much more robust system: in addition to a much more user-friendly post-correction interface, it also includes a version controlled database and API. While on the whole the transition to this new system was a positive development, its timing was not ideal for

this pilot as our work on training data production was delayed several times by minor technical issues involved with the transition to a brand new system. The positive side of this is that we feel confident that most of the technical issues have been discovered and corrected in the pilot.

(4) *Workflow Synchronization:* This pilot involved four phases: (1) initial evaluation of typefaces in *al-Abhath*; (2) training data production; (3) model training and OCR; and, (4) OCR evaluation. At each transfer point there were delays introduced into the workflow because we were not able to synchronize the work schedules of the four different sets of individuals involved in these different phases. This issue is largely the product of the fact that this was a small pilot study and thus the work related to this pilot was not one of the primary job duties of any of the individuals involved. We believe this issue could easily be addressed in any future collaboration through proper planning and sufficient time buyouts for the full range individuals involved in these four phases.

In sum, these issues are all resolvable. Most can be classified as minor "growing pains" that have their origin in the experimental nature of this pilot project.

Works Cited

Kiessling, Benjamin, Matthew Thomas Miller, Maxim Romanov, and Sarah Bowen Savant. "Important New Developments in Arabographic Optical Character Recognition (OCR)." *al-Usur al-Wusta* 25 (2017): 1-13, http://islamichistorycommons.org/mem/wp-content/uploads/sites/55/2017/11/UW-25-Savant-et-al.pdf (accessed 20 January 2019).

Appendix I: OCR Output Page Numbers to *al-Abhath* File Conversion Table

Original File Name	Page Number
54AF8D68-6010-4013-BE4E-37901F2C8966.abhath_23_1-4_ar_0387.tif	67
9B455BE0-EFED-400D-9900-AD7062EBACAD.abhath_27_0_ar_0096.tif	68
7AABE41D-C34E-4B86-9EA6-3926E68B7B10.abhath_16_1_ar_0111.tif	69
F7F0B9C3-35D8-4DC5-BDF6-21C5487DE647.abhath_08_3_ar_0119.tif	70
13B24258-321D-4DFB-A4C6-BE5888B4473B.abhath_21_1_ar_0070.tif	71
BE834161-400D-4B1A-9105-6DB9086B9956.abhath_48-49_0_ar_0097.tif	72
9E008905-D2D1-4272-8A72-9E200D23F323.abhath_27_0_ar_0130.tif	73
52A8D50E-8E1D-478F-B26B-9703ADD55EE3.abhath_23_1-4_ar_0069.tif	74
B611D45D-B8A0-485C-AA6A-8FA59CF43561.abhath_08_3_ar_0013.tif	75
9FDB016F-56F9-4FAE-80C7-82C8AB87B2AD.abhath_27_0_ar_0077.tif	76
75400A8C-A605-4EB2-A8AB-C131FA878B3D.abhath_21_1_ar_0012.tif	77
E9BC01E1-24BF-43C2-8316-AE979DCC9FE4.abhath_23_1-4_ar_0052.tif	78
215DB4D3-834A-4474-870E-3AFC906F418E.abhath_23_1-4_ar_0113.tif	79
C63810B6-9E76-491B-B758-DE9E25C946BE.abhath_27_0_ar_0162.tif	80
0B5E6713-077D-485C-BE0B-F675F0334F20.abhath_23_1-4_ar_0424.tif	81
938F27CB-7792-4E63-AD84-4D5DFA57E42C.abhath_31_0_ar_0025.tif	82
21C4E1A6-4098-4D99-8B26-85A2F4C5383C.abhath_16_1_ar_0136.tif	83
6C73D7A6-F702-4423-AFE5-B6B9A4B61331.abhath_23_1-4_ar_0362.tif	84

D588C725-6C18-48B8-B660-AC80305F43D2.abhath_08_3_ar_0034.tif	85
38C94326-AB19-4172-BEE2-324283DBD40E.abhath_23_1-4_ar_0123.tif	86
0D264DAE-6E59-48A8-BCAB-9448EAC439C4.abhath_08_3_ar_0093.tif	87
D9577746-3D53-4563-B54A-6C04CDA2C376.abhath_23_1-4_ar_0097.tif	88
FED4C1C3-9F9C-48B7-AB7D-D64F45C209DC.abhath_16_1_ar_0032.tif	89
8479DA97-951C-454B-8E6D-6D20EA1A6D20.abhath_34_0_ar_0036.tif	90
E5B34AEE-F50B-45AC-9EE4-F61089129225.abhath_48-49_0_ar_0026.tif	91
EECC8DAE-DFF1-437A-B569-5109CBB2F67A.abhath_16_1_ar_0154.tif	92
74AF23AC-77DF-4270-8153-2E37635BC809.abhath_48-49_0_ar_0034.tif	93
74A88D5C-FC6F-4377-A76B-F294C998AE4B.abhath_16_1_ar_0195.tif	94
265EB8CF-6988-40F1-9665-781515833FA2.abhath_21_1_ar_0049.tif	95
351B3232-83E2-4ABF-9798-11A361D06F3A.abhath_27_0_ar_0131.tif	96
313FBD7A-6DE1-49AA-96BD-0D71B556488C.abhath_27_0_ar_0145.tif	97
DEE64490-DA8D-48F6-B8CC-8DB6D511A71F.abhath_16_1_ar_0039.tif	98
C29FA1D6-4DBE-4E7E-AC72-30835EA0BB59.abhath_48-49_0_ar_0005.tif	99
5259365A-52BD-4251-893C-B792E536BCC9.abhath_08_3_ar_0049.tif	100
E2AD4944-9954-4F47-891F-757EFEB0960A.abhath_27_0_ar_0173.tif	101
91A4690E-F106-4F2C-8DF4-FC999ABA6236.abhath_27_0_ar_0085.tif	102
3F95340D-BF77-4E32-BC08-46497681B320.abhath_08_3_ar_0068.tif	103
FAB0AFA3-830B-4A67-BD8A-4E3C49573E1E.abhath_31_0_ar_0078.tif	104
A3727497-D77C-4FA3-9DCD-78D1DEFA2217.abhath_31_0_ar_0046.tif	105

F3843CC4-C050-4CEC-AD3E-7D09085CE469.abhath_08_3_ar_0122.tif	106
4C1503E2-2E23-4404-A3E2-09C119C23F6A.abhath_16_1_ar_0164.tif	107
C3CCE20A-DDEF-4B02-ABCA-2BCF388BEFC5.abhath_23_1-4_ar_0071.tif	108
A4A9FD30-294A-4468-8976-32FA751E966E.abhath_08_3_ar_0059.tif	109
61A72B88-4833-4BF9-8D09-AFC4A1880BC7.abhath_16_1_ar_0174.tif	110
6C8CAA29-A359-41E7-9936-449B71030D94.abhath_23_1-4_ar_0412.tif	111
A6511FE1-F498-410B-9803-C92A873081FC.abhath_34_0_ar_0031.tif	112
2DE0888B-55F7-4DBC-831D-895514959661.abhath_31_0_ar_0018.tif	113
4BFD1D91-EB86-4119-A3EB-1D0A06188F2A.abhath_16_1_ar_0028.tif	114
64B39822-A827-439A-843D-6CCFA0772E00.abhath_08_3_ar_0045.tif	115
084E1CE9-0429-4E7A-A0A1-7EC263E6680A.abhath_23_1-4_ar_0098.tif	116
2E8CF085-8C5A-4011-9253-0428B6E1B927.abhath_23_1-4_ar_0050.tif	117

Appendix B:

Arabic Journals Supplement Version 1.0 to the JSTOR Journals General Metadata Guidelines Version 1.0. This document is a companion document to the *JSTOR Journals General Metadata Guidelines*, a JSTOR adaptation of the JATS specification.

The *JSTOR Journals General Metadata Guidelines* document can be found at: **https://about.jstor.org/wp-content/uploads/2019/03/JSTOR_Journals_GMG_v.1.0.pdf**

Arabic Journals Supplement

Version 1.0

to the

JSTOR Journals General Metadata Guidelines Version 1.0

Last updated: 28 March 2019

Table of Contents

These guidelines have been made possible in part by the National Endowment for the Humanities: Exploring the human endeavor. Any views, findings, conclusions, or recommendations expressed in these guidelines, do not necessarily represent those of the National Endowment for the Humanities.

The JSTOR Journals General Metadata Guidelines is based in part on the NISO Journal Article Tag Suite, or JATS, standard (ANSI/NISO Z39.96-2015, JATS version 1.1). The JATS standard (ANSI/NISO Z39.96-2015, JATS version 1.1) carries the following copyright statement:

Revision History

Version 1.0, last updated 28 March, 2019

Introduction

Relationship to Journals GMG v1.0

This document is to be used in conjunction with *Journals GMG v1.0* to process journals signed as part of a special Arabic Journals Project.

How to Use These Guidelines

This document includes only the element tables and general sections from the GMG that contain at least one indexing instruction specific to Arabic journals. In addition, it only includes the rules that contain special instructions for processing Arabic journals. For all other rules, element tables, and general sections, refer to *Journals GMG v1.0*.

- A rule that differs from the same-numbered rule in *Journals GMG v1.0* is indicated by gray shading of the rule number cell. When processing an Arabic journal issue, apply a rule marked in this manner *instead of* the same-numbered rule in *Journals GMG v1.0*.
    - For example:

9.28		If there is an untitled introductory article associated with an article group, supply an article title as follows: • If the article is in Arabic, index <article-title> [مقدمة] • If the article is in English, index <article-title> [Introduction]

- A rule which is unique to this document (i.e., not in *Journals GMG v1.0*) is indicated by TWO features: 1) the rule number cell is shaded green, and 2) the rule number is in the format *number.letter*. When processing an Arabic journal issue, apply rules marked in this manner *in addition to* rules in *Journals GMG v1.0.* (By using unique designations for rules that appear only in this document, consistency in rule numbers can be maintained between the two documents.)
    - For example, <pub-date> contains four rows (a heading plus three rules) that apply only to Arabic Journals, numbered **88.a** through **88.d**, situated after GMG rule 88.17:

88.17		…
88.a		**Arabic Journals: Generating a Gregorian Date**
88.b		If a Gregorian-calendar date is present in the source, …
88.c		If a Gregorian-calendar date is not present in the source, …

88.d		The vendor may use an online date converter …

JSTOR will notify the digitization vendor which journals are to be processed using the *Arabic Journals Supplement*. Journals in the JSTOR Archive Collections that contain some Arabic content but are not signed as part of the Arabic Journals Project are to be processed using only *Journals GMG v1.0*, unless otherwise directed by JSTOR.

9. <article-title> - Article Title

9	**Element**	<article-title>
9.1	**Descriptor**	Article Title
…		…
9.28		If there is an untitled introductory article associated with an article group, supply an article title as follows: - If the article is in Arabic, index <article-title> [مقدمة] - If the article is in English, index <article-title> [Introduction]
…		…

20. <copyright-statement> - Copyright Statement of Issue or Article

20	**Element**	<copyright-statement>
20.1	**Descriptor**	Copyright Statement of Issue or Article
…		…
20.15		…
20.a		If a copyright statement is present in more than one language, capture each language statement in a separate <copyright-statement>. Example: <permissions> <copyright-statement> © حقوق النشر محفوظة لقسم الأدب الإنجليزي ٢٠٠٥، والمقارن وقسم النشر بالجامعة الأمريكية بالقاهرة </copyright-statement> <copyright-statement>© 2005, Department of English and Comparative Literature, the American University in Cairo</copyright-statement> </permissions>
…		…

22. <cover-image> - Cover Image

22	**Element**	<cover-image>
22.1	**Descriptor**	Cover Image
…		…
22.10	**Location in source**	If an issue has an Arabic front cover at one end and an English front cover at the other end, so that the issue can be read from either direction, use only the Arabic front cover for <cover-image>.
…		…

36. <gmg-version> - General Metadata Guidelines (GMG) Version

36	**Element**	<gmg-version>
36.1	**Descriptor**	General Metadata Guidelines (GMG) Version
…		…
36.9	**Format required**	Enter the version of the metadata guidelines and the version of the Arabic Journals Supplement used in the production of the issue's XML files in the format "Journals GMG X.X + Arabic Journals Supplement X.X", substituting the actual version number of each document for "X.X". Example: <gmg-version>Journals GMG 1.0 + Arabic Journals Supplement 1.0</gmg-version>
…		…

44. <issue-page-range> - Issue Page Range

44	**Element**	<issue-page-range>
44.1	**Descriptor**	Issue Page Range
...		...
44.28		If an issue contains content in both Arabic and English, and the entire issue is paginated in one sequence, that is the only sequence entered in <issue-page-range>. Example: For an issue that has Arabic content at the front of the issue, and English content that begins at the end of the physical issue and progresses inward so that it ends in the middle of the issue: Scanning order: [Arabic] nil, 1, 2, 3, 4... 100, [English] nil, 101, 102, 103, ... 150 Capture the issue page range as: <issue-page-range> 1-150 Example: For an issue that has Arabic content at the front of the issue, and English content that begins at the end of the physical issue and progresses inward so that it ends in the middle of the issue: Scanning order: [Arabic] nil, 1, 2, 3, 4, ... 78, [English] nil, 122, 121, 120, ... 80, 79 Capture the issue page range as: <issue-page-range> 1-122

44.29		If an issue contains content in both Arabic and English, and each language section has its own pagination sequence, enter both sequences. Capture each sequence so that the starting page number is first, then a hyphen, then the ending page number. In an issue that contains some content read left-to-right and some content read right-to-left, the page numbers of some articles may progress from higher to lower. However that is NOT necessarily the order that the pagination should be input. Capture each page range in the logical order that corresponds to the character system. • If the page numbers are standard Arabic numerals (e.g., 99, 98, 97, 96, 95) or any language/numeral system read left-to-right (Roman numerals, Greek, Cyrillic, etc.), index the range in left-to-right order, or lowest number-highest number (e.g., 95-99). • If the page numbers are Arabic, Arabic script, or any language/numeral system read right-to-left, index the range in right-to-left order. Example: For an issue that has Arabic content at the front of the issue, and English content that begins at the end of the physical issue and progresses inward so that it ends in the middle of the issue: Scanning order: [Arabic] nil, 1, 2, 3, … 48, [English] nil, 1, 2, 3, … 27 Capture the issue page range as: <pagerange> 1-48, 1-27 Example: Same as above example, but the Arabic section contains Arabic-script page numbers:

		Scanning order: [Arabic] nil,٤٨, ... ٣, ٢, ١,[English] nil, 1, 2, 3, ... 27 Capture the issue page range as: <pagerange> ١ -٤٨, 1-27 (The Arabic page range would be understood by readers of Arabic as "1-48" because they would read the range of numbers from right to left.)
...		...

47. <issue-title> - Issue Title

47	**Element**	<issue-title>
47.1	**Descriptor**	Issue Title
…		…
47.27		If an issue title is in more than one language, capture all language versions as they appear in the source for capitalization, punctuation, and spacing. • If punctuation is not present between translated issue titles--for instance, the translation is on the next line or is located elsewhere in the issue--index “space, slash, space” between translated issue titles. • For Arabic journals, the format (in a right-to-left reading orientation) is "Translation / PrimaryLanguage" or "Transliteration / PrimaryLanguage". Example: On Arabic cover: الشعرية الظاهرة On English cover: The Lyrical Phenomenon Index as: <issue-title> The Lyrical Phenomenon / الشعرية الظاهرة
…		…

71. <name> - Personal Name

71	**Element**	<name>
71.1	**Descriptor**	Personal Name
...		...
71.8	**Occurrence**	Page Scan, PDF: • One <name> per <contrib> for a contributor name in Latin, Arabic, Hebrew, or Cyrillic characters with a discernible surname. • One or more <name> per <product> for contributor names in Latin, Arabic, Hebrew, or Cyrillic characters with a discernible surname, only when <product> is NOT in citation format in the source (see <product> for instructions). Note exception to first point above: One or more <name> per <name-alternatives> inside <contrib> when multiple versions of a personal contributor name are listed for an article, and at least one version is in Latin, Arabic, Hebrew, or Cyrillic characters and has a discernible surname. Full-Text: Preserve <name> if present, provided it complies with the JATS model.
...		...

77. <page> - Individual Page in an Issue

77	**Element**	<page>
77.1	**Descriptor**	Individual Page in an Issue
...		...
77.17		The numbers in this attribute correspond to the scanned pages in the issue. Scan the issue in reading order. Assign p-1 to the first page that a native speaker would read and continue numbering pages upwards in reading order. For an issue that contains separate Arabic and English sections: • Scan each section in reading order. • Scan the Arabic section first. Example: If an issue contains 85 contiguous scanned pages, then the sequence of "id" numbers is 1 2 3 4 ... 82 83 84 85.
...		...
77.50	**Internal Process Notes**	
77.51		Per Journals GMG 1.0, pages must be scanned in physical order. The reason for this requirement is because JSTOR staff may need to know how pages were organized in the physical issue. If pages are scanned in some other order, the original sequence of pages cannot be determined. However, for journals digitized as part of the Arabic Journals Project, JSTOR has made an exception to the requirement to scan pages in physical order.
...		...

87. <product> - Reviewed Works

87	**Element**	<product>
87.1	**Descriptor**	Reviewed Works
…		…
87.15		Citations of reviewed works may be found in the following locations: • At the beginning of the review article • In a footnote at the bottom of the first page, which is referenced by a symbol (e.g., an asterisk) at the end of the article title • In a box, either off to the side or at the beginning of the review text • In or near a thumbnail cover image of the reviewed work • Labeled (e.g., "Books Reviewed" or "Review of") at the beginning or end of a review or group of reviews **Additionally, for Arabic journals:** rate and distinct Arabic-script and Latin-script citations for the same reviewed work appear in an issue, mark up/capture each one in a separate <product>. The order of preference for separate translated/transliterated product information is: 1) with the article 2) in a table of contents 3) with an abstract that is physically separated from the article
87.16		Occasionally, <product> information may appear listed with review articles in the TOC. If what looks to be <product> appears only in the TOC and does not appear at the article level in the locations indicated above, submit an Indexing Query in JIRA to the JSTOR librarians to determine if <product> should be captured.

🟩		● NOTE: For Arabic journals, this rule does not pertain to an article that has product information at the article level AND ALSO has separate translated/transliterated product information in a TOC. In this case, do not submit a query; capture the product information from the TOC as additional <product> for the article, as instructed in the preceding rule.
…		…

88. <pub-date> - Publication Date as Numerical Values

88	**Element**	<pub-date>
88.1	**Descriptor**	Publication Date as Numerical Values
…		…
88.9	**Format required**	Index the Gregorian-calendar date using standard Arabic numerals. If a Gregorian-calendar date is not present in the source, the Islamic/Hijri-calendar date must be converted to a Gregorian-calendar date. See instructions below under the heading "Arabic Journals: Generating a Gregorian Date".
…		…
88.17		…
88.a		**Arabic Journals: Generating a Gregorian Date**
88.b		If a Gregorian-calendar date is present in the source, base the <pub-date> on the Gregorian date.
88.c		If a Gregorian-calendar date is not present in the source, use the Islamic/Hijri-calendar date in the source to generate the <pub-date> values according to the following rules:

Islamic/Hijri-Calendar Date in Source:	**Conversion Instructions**
Specific date (Day, Month, and Year)	Convert to the corresponding specific Gregorian date (there is a one-to-one correspondence).
Month and Year	Assume the issue was published on the first day of the specified Islamic month. Convert that specific Islamic date to the corresponding Gregorian date.

Season or Quarter and Year	Submit an Indexing Query in JIRA to the JSTOR librarians.
Year	Convert the first day of the month Muharram of the Islamic year in the source to a Gregorian date. If the Gregorian date falls in January, capture the year in that date in <pub-date> (with <day> and <month> values of '1'). If the Gregorian date falls in any other month (February to December), capture the year in that date in the first <pub-date> (with <day> and <month> values of '1') and capture the subsequent year in a second <pub-date> (with <day> and <month> values of '1'). See examples in Appendix 1 at the end of this document.
Range of dates	Convert each stated portion of the date range to the corresponding Gregorian date using the instructions above in this table, and index each Gregorian date in a separate <pub-date>.

88.d

The vendor may use an online date converter to convert an Islamic/Hijri-calendar date to a Gregorian-calendar date or to convert a Gregorian year in the form of Arabic-script numerals to standard Arabic numerals. Examples of online converters include:

http://dateconverter.net/arabic/

https://www.linktoislam.net/islamic-calendar/hijri-date-converter/

http://www.islamicity.org/hijri-gregorian-converter/

http://www.icoproject.org/conv?l=en

http://www.al-islam.com/Loader.aspx?pageid=918

https://www.islamicfinder.org/islamic-date-converter/

…		…

103. <string-date> - Date as String

<table>
<tr><td><b>103</b></td><td><b>Element</b></td><td>&lt;string-date&gt;</td></tr>
<tr><td><b>103.1</b></td><td><b>Descriptor</b></td><td>Date as String</td></tr>
<tr><td><b>…</b></td><td></td><td>…</td></tr>
<tr><td><b>103.15</b></td><td></td><td>…</td></tr>
<tr><td><b>103.a</b></td><td></td><td>
<p>Date information in more than one character set:</p>
<p>If the date is presented in Arabic-script characters, capture the Arabic-script date only, and ignore a Latin-script translation/transliteration of the date. (The Arabic-script date may contain standard Arabic numerals.) If the date is not presented in Arabic characters, capture it in the language(s) presented.</p>
<ul>
<li>If an Islamic/Hijri-calendar date and a Gregorian-calendar date both appear on the issue in Arabic script (which may include standard Arabic numerals), capture both dates. If there is no date in Arabic characters on the issue, and dates from both calendar systems appear on the issue in Latin script, capture both dates. In either case, if no punctuation is present between the two dates on the source, place "space, slash, space" between them.</li>
<li>If the issue date consists of a year only, and the year appears on the issue in both Arabic script and standard Arabic numerals, capture both years. If no punctuation is present between the two dates on the source, place "space, slash, space" between them.</li>
</ul>
<table>
<tr><th>Date on Arabic cover:</th><th>Date on English cover:</th><th>Index in <string-date&gt;:</th></tr>
<tr><td>١٩٤٨</td><td>1948</td><td>١٩٤٨ / 1948</td></tr>
<tr><td>حزيران 2014</td><td>June 2014</td><td>حزيران 2014</td></tr>
<tr><td>ذو الحجة 1398 / نوفمبر 1978</td><td>November 1978</td><td>ذو الحجة 1398 / نوفمبر 1978</td></tr>
</table>
</td></tr>
</table>

<table>
<tr>
<td></td>
<td></td>
<td>
<table>
<tr><td>١ جمادى الأول ١٤٢٩</td><td>1 Jumada al-Awwal 1429 / May 6 2008</td><td>١ جمادى الأول ١٤٢٩</td></tr>
<tr><td>شباط - ١٩٨٨ الأول كانون ١٩٨٩</td><td>Kānūn al-Awwal 1988-Shbāṭ 1989 / Dec. 1988 - Feb. 1989</td><td>شباط - ١٩٨٨ الأول كانون ١٩٨٩</td></tr>
<tr><td>فبراير - ١٩٨٨ ديسمبر ١٩٨٩</td><td>Dīsambir 1988 – Fibrāyir 1989</td><td>فبراير - ١٩٨٨ ديسمبر ١٩٨٩</td></tr>
<tr><td>Kānūn al-Thānī 2003 - Tammūz 2004 / Jan. 2003-July 2004</td><td>Kānūn al-Thānī 2003 - Tammūz 2004 / Jan. 2003-July 2004</td><td>Kānūn al-Thānī 2003 - Tammūz 2004 / Jan. 2003 - July 2004</td></tr>
<tr><td>Muharram-Safar 1409 / Sept. 1988</td><td>Muharram-Safar 1409 / Sept. 1988</td><td>Muharram-Safar 1409 / Sept. 1988</td></tr>
</table>
</td>
</tr>
<tr>
<td>…</td>
<td></td>
<td>…</td>
</tr>
</table>

104. <string-issue> - Issue Number(s) as String

104	**Element**	<string-issue>
104.1	**Descriptor**	Issue Number(s) as String
…		…
104.9	**Format required**	Index one or more Arabic-script numbers or standard Arabic numbers in <string-issue>. A number may be followed by a letter. Punctuation and other characters are allowed. Omit a label that precedes or follows the issue number.
…		…
104.14		…
104.a		If the issue number is presented in Arabic characters, capture the Arabic-script issue number only, and ignore an issue number presented in any other numeral system. If the issue number is not presented in Arabic characters, capture it as a standard Arabic numeral.
104.15		If the issue number is a Roman numeral, convert it to the corresponding standard Arabic numeral. Example: Index issue number "XXVI" as <string-issue>26</string-issue>.
104.16		If the issue number is spelled out as an Arabic word, convert it to the corresponding Arabic-script numeral. Example:

🟩		Index the word الأول, أول or واحد as <string-issue>١</string-issue>. If an Arabic-script issue number is not present in the source, and the issue number is spelled out as an English word, convert it to the corresponding standard Arabic numeral. Example: Index the word "First" or "One" as <string-issue>1</string-issue>.
…		…

106. <string-name> - Contributor Personal Name: Unstructured

106	**Element**	<string-name>
106.1	**Descriptor**	Contributor Personal Name: Unstructured
…		…
106.1 6		In either <contrib> or <product>, use <string-name> for unstructured personal contributor names that cannot be parsed because they do not have a discernible surname; therefore, use <string-name> in the following situations: • Contributor information consists entirely of a single name which is not a surname. Example: <string-name>Mohammed</string-name> • Contributor information consists of only initials. Example: <string-name>J. L. S.</string-name> • Contributor information consists of an honorific or courtesy title followed by a name which is not a surname. Example: <string-name>Prince Charles</string-name> <string-name>Pope Sylvester II</string-name> <string-name>Saint Boniface</string-name> <string-name>Sister Mary Agnes</string-name> <string-name>Brother James</string-name>

		• Contributor information consists of a title only, with no accompanying name information. Example: <string-name>Duke of Essex</string-name> <string-name>Prime Minister of Her Majesty's Government</string-name> • Contributor information is not a person’s name or the name of an organization. Example: <string-name>Anonymous</string-name> <string-name>The Editors</string-name> <string-name>A Concerned Citizen</string-name> • Contributor information is a single name, and it is not possible to determine from the context whether it is a person's surname. Example: <string-name>Közi</string-name> • Contributor information is in a character set other than Latin, Arabic, Hebrew, or Cyrillic, e.g. Chinese, Japanese, etc. Example: <string-name>朱維理</string-name>
…		…

107. <string-volume> - Volume Number(s) as String

107	**Element**	<string-volume>
107.1	**Descriptor**	Volume Number(s) as String
…		…
107.9	**Format required**	Index one or more Arabic-script numbers or standard Arabic numbers in <string-volume>. A number may be followed by a letter. Punctuation and other characters are allowed. Omit a label that precedes or follows the volume number.
…		…
107.13		…
107.a		If the volume number is in Arabic characters, capture the Arabic-script volume number only, and ignore a volume number in any other numeral system. If the volume number is not in Arabic characters, capture it as a standard Arabic numeral.
107.14		If the volume number is a Roman numeral, convert it to the corresponding standard Arabic numeral. Example: Index volume number "XXVI" as <string-volume>26</string-volume>.
107.15		If the volume number is spelled out as an Arabic word, convert it to the corresponding Arabic-script numeral. Example: Index the word الأول, أول or واحد as <string-volume>١</string-volume>.

🟩		If an Arabic-script volume number is not present in the source, and the volume number is spelled out as an English word, convert it to the corresponding standard Arabic numeral. Example: Index the word "First" or "One" as <string-volume>1</string-volume>.
…		…

116. <surname> - Surname

116	**Element**	<surname>
116.1	**Descriptor**	Surname
…		…
116.20		Some names of Arabic origin contain the prefix إبن or ابن or "ibn" (meaning "son of"). This prefix may appear more than once in a person’s name and may or may not be capitalized in the source. Capture from the last (or only) instance of إبن or ابن or "ibn" to the end of the name in <surname>. Examples: Name in source: إبراهيم إبن محمد Capture as: <surname>إبراهيم إبن</surname> <given-names>محمد</given-names> Name in source: العزيز عبد ابن فيصل ابن خالد Capture as: <surname>العزيز عبد ابن</surname> <given-names>فيصل ابن خالد</given-names> Name in source: Ibn Hazm Capture as: <surname>Ibn Hazm</surname>

		Name in source: Ibn Rashiq al-Qayrawani Capture as: <surname>Ibn Rashiq al-Qayrawani</surname> Name in source: Muhammed ibn Badr Jajarmi Capture as: <surname>ibn Badr Jajarmi</surname> <given-names>Muhammed</given-names> Name in source: Ahmad Ibn ‘Abd Al-Halim Ibn Taymiyyah Capture as: <surname>Ibn Taymiyyah</surname> <given-names>Ahmad Ibn ‘Abd Al-Halim</given-names>
116.a		Other surname prefixes besides "ibn" can occur in names of Arabic origin. These are listed in Appendix 2 at the end of this document. A surname prefix may or may not be capitalized in the source.
116.b		If only one surname prefix is present in an Arabic name, index it as part of <surname>. If more than one potential surname prefix is present, apply the appropriate instruction below: ● If two adjacent surname prefixes precede the final part of the name, capture both prefixes as part of <surname>. Example:

		Name in source: الشيخ آل بن محمد Capture as: <surname>الشيخ آل بن</surname> <given-names>محمد</given-names> Example: Name in source: Omar ibn Al Khattab Capture as: <surname>ibn Al Khattab</surname> <given-names>Omar</given-names> ● If a surname prefix is present before the last part of the name, and another potential surname prefix occurs earlier in the name but is NOT adjacent to the last prefix, capture only from the last prefix to the end of the name in <surname>. Capture all parts of the name preceding the last prefix in <given-names>. Example: Name in source: آل العزيز عبد بن عبدالله سعود Capture as: <surname>آل سعود</surname> <given-names>عبدالله عبد بن العزيز</given-names> Example: Name in source: Sabika bint Ibrahim Al Khalifa Capture as: <surname>Al Khalifa</surname> <given-names>Sabika bint Ibrahim</given-names>

		Note that the instructions in this rule override the first Indexing Instruction given in the element table for <surname>. If a name is presented in reverse order, use the instructions in this rule, not the arrangement of the name in the source, to determine which parts of the name to capture in <surname>.
116.c		Special instruction for عبد or "Abd": عبد or "Abd" can be a surname prefix OR a given name. Capture it as part of <surname> only if it falls between the given name(s) and the last part of the name. Example: Name in source, where عبد is not a surname prefix: الطاهر عبد Capture as: <surname>الطاهر</surname> <given-names>عبد</given-names>
116.d		Special instruction for الله, "Allah", or "Ullah": When الله, "Allah", or "Ullah" is present at the end of a person's name, capture also the one-word part of the name immediately preceding it (i.e., the second-to-last part of the name) in <surname>. Example: Name in source: الله حبيب جواد Capture as: <surname>الله حبيب</surname> <given-names>جواد</given-names>

		Example: Name in source: Imad Nasr Allah Capture as: <surname>Nasr Allah</surname> <given-names>Imad</given-names>

117. <title> - Title

117	**Element**	<title>
117.1	**Descriptor**	Title
...		...
117.2 8		If an article group title is in more than one language, capture all language versions as they appear in the source for capitalization, punctuation, and spacing. • If punctuation is not present between translated/transliterated article group titles--for instance, the translation is on the next line or is located elsewhere in the issue--index “space, slash, space” between translated/transliterated article group titles. • For Arabic journals, the format (in a right-to-left reading orientation) is "Translation / PrimaryLanguage" or "Transliteration / PrimaryLanguage". In issues containing Arabic and English sections: • The primary language group heading for articles in the ARABIC section is the Arabic heading printed at the article level or in an Arabic TOC. If a Latin-script translation/transliteration of the heading is available at the article level or in a TOC, capture it as part of <title>. • The primary language group heading for articles in the ENGLISH section is the English heading printed at the article level or in an English TOC. If an Arabic-script translation/transliteration of the heading is available at the article level or in a TOC, capture it as part of <title>.
...		...
117.3 7		Page Scan, PDF: When the source does not provide a title for a group of references, supply a title in square brackets, using one of the terms listed below. If the article is in Arabic, select the most appropriate Arabic

		title. If the article is in English or any other non-Arabic language, select the most appropriate English title. • [الببليوغرافيا] or [Bibliography] -- when the references appear as part of a numbered or alphabetical end-of-article bibliography (i.e., an alphabetical list of cited works). References contained in such bibliographies usually do not include any text in addition to that of the citation itself, so the extent of a reference will correspond exactly with the extent of the citation. • [الهوامش] or [Endnotes] -- when the references include explanatory or analytical text in addition to the citation itself and appear as part of an ordered list at the end of an article. The numbers/letters/symbols of the ordered list refer to the order in which the text the endnote refers to appears in the body of the article. • [الحواشي] or [Footnotes] -- when the references appear at the end of pages throughout the article. The numbers/letters/symbols of the ordered list refer to the order in which the text the footnote refers to appears in the body of the article, or sometimes to the order in which they appear on an individual page. Footnotes often include explanatory or analytical text in addition to the citation itself.
…		…
117.41		Full-Text: When the source does not provide a title for a group of references that apply to the article as a whole, supply a title in square brackets, using one of the terms listed below. If the article is in Arabic, select the most appropriate Arabic title. If the article is in English or any other non-Arabic language, select the most appropriate English title. • [الببليوغرافيا] or [Bibliography] -- when the references appear as part of a numbered or alphabetical end-of-article bibliography (i.e., an alphabetical list of cited works). References contained in such bibliographies usually do not include any text in addition to that of the citation itself, so the extent of a reference will correspond exactly with the extent of the citation.

		• [الهوامش] or [Notes] -- when the references include explanatory or analytical text in addition to the citation itself and appear as part of an ordered list at the end of an article. The numbers/letters/symbols of the ordered list refer to the order in which the text the endnote refers to appears in the body of the article.

124. <trans-title> - Translated Title

124	**Element**	<trans-title>
124.1	**Descriptor**	Translated Title
...		...
124.10	**Location in source**	See Indexing Instructions.
...		...
124.15		Capture each translated version of an article title (and subtitle if applicable) present in the issue in a separate <trans-title-group>.
124.16		Capture a translation/transliteration of the article title if present in the issue. The order of preference for a translated/transliterated article title is: 1) at the head of the article (where it may appear with a translated abstract) 2) in a table of contents 3) with an abstract that is physically separated from the article
...		...

135. Contributor Information in Page Scan and PDF Source

135	**Section Title**	Contributor Information in Page Scan and PDF Source
…		…
135.8		Do not index as part of a contributor's name the cross or dagger symbol placed after the name to indicate that the person is deceased. ● For Arabic names, do not capture phrases such as رحمه الله, رحمها الله, رحمه الله تعالى, رحمها الله تعالى, "Rahamahu Allah", or "Rahamaha Allah" placed after the name to indicate that the person is deceased. Example: Name in source: داني شمعون رحمه الله Capture as: <surname>شمعون</surname> <given-names>داني</given-names>
…		…
135.10		For contributor names in characters other than Latin, Arabic, Hebrew, or Cyrillic, use <string-name>.
…		…
135.14		**Contributor to Article**
135.15		**Location in Source Instructions:**

		Page Scan, PDF: Contributor information for an article is usually listed on the initial page of the article, but it can also appear at the end of the article, in an introduction to the article, at the beginning or end of sections within an item, and/or in the table of contents. PDF: If contributor information is not present in PDF source for articles of type "research-article", "review-essay" or "book-review", look for contributors in publisher-provided XML file(s), if available. If contributor(s) are found there, submit an Indexing Query in JIRA for a decision on capturing them, and do not look further. If contributor information is not found in publisher-provided XML file(s) (or if such files do not exist), then look for contributor information on the publisher's website. If contributor(s) are found there, submit an Indexing Query in JIRA for a decision on capturing them. **Additionally, for Arabic journals (Page Scan, PDF):** Capture all article contributors in both Arabic characters and Latin characters from any of the locations in the list above labeled "Page Scan, PDF". If Arabic-script and Latin-script versions of a particular name are listed in these locations, capture both versions in separate <name>, <collab>, and/or <string-name> elements within <name-alternatives> or <collab-alternatives>. The order of preference for translated/transliterated article author information is: 1) with the article 2) in an issue table of contents 3) with an abstract that is physically separated from the article
…		…

136. Date Information for the Issue Being Processed

136	Section Title	Date Information for the Issue Being Processed
…		…
136.8		Submit an Indexing Query in JIRA to the JSTOR librarians if any of the following publication date problems are encountered: • If publication date is not available. • If there are publication date oddities, irregularities, or misprints. • If there are discrepancies between prominent sources of date information. • If the issue has both a coverage date and a publication date. • If the only date available is a copyright date. For regular Archive Collections journals, the vendor is required to submit a query when date information in the source is in more than one language. However, for the Arabic Journals Project, a query is NOT required because this situation is covered by instructions in the <pub-date> and <string-date> element tables above.
…		…

140. Instructions for Handling Illustrations in Page Scan and PDF Source

140	**Section Title**	Instructions for Handling Illustrations in Page Scan and PDF Source
…		…
140.10		**When illustrations are not related to an article:**
140.11		Create an <article> for each illustration or for each grouping of illustrations. • Use article-type "misc". • Index <article-title> as "[صورة إيضاحية]" for an individual illustration. • Index <article-title> as "[رسوم توضيحية]" for a group of illustrations. • Index a caption if present.
…		…

141. Issue Front Matter and Back Matter in Page Scan and PDF Source

141	**Section Title**	Issue Front Matter and Back Matter in Page Scan and PDF Source
…		…
141.17		Page Scan: Place in Front Matter any pages of non-substantive content which appear in the issue before or within the last article. PDF: Place in Front Matter any pages of non-substantive content which appear in the issue before the first article. • Special Case: If an issue has an Arabic front cover at one end and an English front cover at the other end, so that the issue can be read from either direction, index the English cover, English TOC, and any other non-substantive pages at the front of the English section in Front Matter, not in Back Matter. Within Front Matter, index the pages at the front of the Arabic section first, followed by the pages at the front of the English section. Index pages of the English section in order starting with the front cover, followed by the inside front cover, and so on.
…		…

141. 20		For an issue that is entirely or partially in Arabic, index the article titles of these two articles exactly as follows: • For Front Matter: <article-title> الأمامية المادة • For Back Matter: <article-title> المؤخرة المادة For an issue that is entirely in English or any other non-Arabic language, index the article titles of these two articles exactly as follows: • <article-title> Front Matter • <article-title> Back Matter
…		…

Appendix 1: Examples of Conversion from Islamic/Hijri Year to Gregorian Year(s)

Example:

Year in source: 1361 of the Islamic/Hijri calendar

Conversion: 1 Muharram 1361 = January 18, 1942 (i.e., Islamic/Hijri year 1361 corresponds roughly to Gregorian year 1942)

<pub-date>

<day>**1**</day>

<month>**1**</month>

<year>**1942**</year>

</pub-date>

Example:

Year in source: 1384 of the Islamic/Hijri calendar

Conversion: 1 Muharram 1384 = May 12, 1964 (i.e., Islamic/Hijri year 1384 corresponds roughly to Gregorian year range 1964/1965)

<pub-date>

<day>**1**</day>

<month>**1**</month>

<year>**1964**</year>

</pub-date>

<pub-date>

<day>**1**</day>

<month>**1**</month>

<year>**1965**</year>

```
</pub-date>
```

Appendix 2: Surname Prefixes

Arabic-Character Surname Prefixes

Surname Prefix	Transliteration	Example of Contributor Name in Source	Indexed in <surname>
أبو	Abu or Abou	عاصي أبو بيير	عاصي أبو
أبي	Abi	زيد أبي بسام	زيد أبي
آل	Al, Aal or Āl	عمران آل مصطفى	عمران آل
أم	Umm	كلثوم أم	كلثوم أم
بن	Bin	العزيز عبد بن طلال بن الوليد	العزيز عبد بن
بنت	Bint	العزيز عبد بنت مريم	العزيز عبد بنت
بو	Bo or Bou	خليل بو جوزيف	خليل بو
عبد	Abd	الله عبد عمر	الله عبد

Latin-Character Surname Prefixes

Surname Prefix	Example of Contributor Name in Source	Indexed in <surname>
Abd	Faruq Abd Allah	Abd Allah
Abdel	Walid Abdel Fatah	Abdel Fatah
Abdul	Munir Abdul Rahman	Abdul Rahman
Abdur	Shareef Abdur Rahman	Abdur Rahman
Abi	Hiba Abi Nasr	Abi Nasr
Abou	Bassam Abou Zeid	Abou Zeid

Abu	Souheil Abu Jamra	Abu Jamra
Al	Sabika Al Khalifa	Al Khalifa
Āl	Jalal Āl Hamad	Āl Hamad
Bin	Mohammad bin Salman	bin Salman
Bint	Hind bint Abi Umayya	bint Abi Umayya
Bo	Adel Bo Nassif	Bo Nassif
Bou	Wajdi Bou Khalil	Bou Khalil
Umm	Umm Ali	Umm Ali